One Algorithm to Rule Them All

Johanna Voznak, University of Delaware

Evan Bongiovanni[a], Nick Brown[b], Calum Buchanan[c], Noah Chicoine[d], Brooks Emerick[e], Adebayo Emmanuel[f], Caroline Hammond[g], Kayode Oluwasegun[h], Olivia Pomerenk[i], Luis Schneegans[j], Hemaho Taboe[k]

[a]Portsmith, RI, [b]Penn State University, [c]University of Vermont, [d]Northeastern University, [e]Kutztown University, [f]University of Delaware, [g]Dartmouth College, [h]Drexel University, [i]New York University, [j]Stanford University, [k]University of Florida,

ARTICLE HISTORY
Compiled August 12, 2024

1. Introduction

The Graduate College at the University of Delaware has implemented the Grad LEAP program over the past several years to connect current graduate students to alumni mentors. The program aims to identify willing mentors and match them to similar mentees under a certain set of criteria that includes cultural background, academic disciplines, career interests, and specific program goals among others. With strong connections in place, the Graduate College hopes the mentor will provide guidance and support as the mentee navigates through their graduate career.

Currently, the Graduate College contracts with an educational technology vendor to optimize the matching process between mentees and mentors. The process is achieved by analyzing data gathered from mentee and mentor surveys. The Graduate College has analyzed the survey data and and the matching results from Cohort 3, a group of 293 possible mentees and 150 possible mentors. However, the resulting matches have been less than ideal. Post-surveys reveal that only about 30% of matchings were considered "good" with many having to be re-matched. Furthermore, some mentee-mentor pairs do not match on the most important elements such as academic discipline or career interests.

The Graduate College seeks to achieve the following goals from the Mathematical Problems in Industry workshop:

- Develop a more robust, user-friendly algorithm that maximizes "good" matches and is compatible with survey data formatted from Google Forms.
- Review the current mentee and mentor surveys to determine if useful data is being collected and make revisions where necessary.
- Generate a confidence score with appropriate rationale so that potential matches can be reviewed.

In the sections that follow, we develop a sound framework for determining optimal matches. We achieve this in two phases: First, we use the availabile data from the current survey to construct two mathematical formulations. We use this available data to validate the model, which provides proof of concept. Second, we revise the current survey to be structured in such a way that our formulations matche mentees and mentors on every important criteria. This includes built-in Likert scales that can be used to effectively weight each criteria. Then, we formulate both mathematical models to conform to the new survey results. We validate the model using synthetic data based on actual data provided for Cohort 3. The final product is an algorithm to be used in future iterations of the graduate mentee-mentor program.

This paper is organized as follows: In Section 2 we define compatibility between any potential mentee-mentor pair using the available survey results from Cohort 3. This definition of compatibility is then used to formulate a feasible linear programming problem (LP) that is solved for Cohort 3 using the Hungarian Method. We then use compatibility to dictate a ranking system for the stable marriage problem (SMP) formulation, which is solved for Cohort 3 using the Gale-Shapley Method. We discuss the results of each model by computing appropriate measures and compare the output of our matching algorithms. In Section 3, we examine the various issues with the current survey format and suggest a new survey with a built-in ranking system to aid in the construction of compatibility. We then construct the LP and SMP formulations using synthetically generated output from the reformatted survey and develop a complete program package that gives optimal pairings for future cohorts. In Section 4, we consider two other alternative solutions using the Jaccard similarity index and a machine learning approach. Finally, in Section 5 we conclude with a discussion.

2. Formulations Based on Existing Survey Data

A solution to this problem exists but requires an appropriate formulation. Indeed, the problem can be solved using either a linear programming (LP) or a stable marriage problem (SMP) approach. We explore both avenues. To achieve an appropriate LP formulation, we must define an objective function and a set of linear constraints on the decision variables. The objective function is dependent on a metric that measures either the *compatibility* of every possible mentee-mentor assignment. To achieve an appropriate SMP formulation, we must have a ranking system in place. That is, each mentee has a preferred ordering of potential mentors. In both cases, we seek a relationship between mentees and mentors that measures the strength of every potential pairing. Before we proceed with solving each respective problem, we construct this relationship by defining a compatibility score. In the subsections that follow, we present the criteria necessary to define compatibility and then we formulate and solve each model.

2.1. Criteria and Weights

We first define the relationship between every possible mentee-mentor pairing by constructing a compatibility matrix, C, of size $m \times n$, where m denotes the number of mentees and n denotes the number of mentors. Here, each element c_{ij}, for $i \in \{1, 2, \dots m\}$ and $j \in \{1, 2, \dots, n\}$, of the matrix C is a numerical measure of compatibility between mentee s_i and mentor t_j. Each element of this matrix is calculated based on criteria explicitly defined from the provided Cohort 3 survey results. The goal is to match an optimal mentor to each mentee. That is, each mentee must be matched, but a mentor need not be matched. Figure 1 shows a schematic of an optimal match with compatibility values shown between each connecting edge.

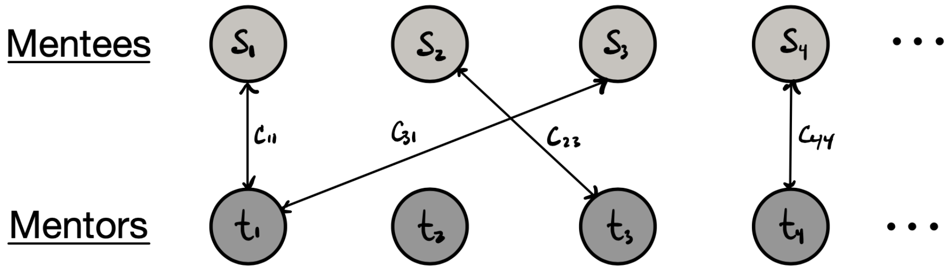


Figure 1. Schematic representation of the m mentees and n mentors with their optimal matches as determined by compatibility score c_{ij}. Every mentee must have a single mentor, but not every mentor is assigned to a mentee. Furthermore, mentors can have more than one mentee depending on each mentor's capacity according to Question 7 of the survey.

We identify eight questions from both the mentee and mentor surveys that are linked. By analyzing each participant's response, we are able to make assumptions into the similarities of a particular mentee and mentor pairing. Therefore, we associate each linked question to a particular criterion to be used to create a compatibility score. The criteria are detailed in Table 1.

The survey questions associated to each criteria typically have a drop-down list of possible responses. A majority of the questions also have the option for the participant to select multiple options in the drop-down list. Hence, we examine the given answer(s) to each linked question and observe the number of matches that occur. It is known that a set of particular survey questions are more important to the mentee than others. Indeed, Question 9 on the mentee survey asks directly what attributes of a potential mentor are most important in determining whether or not that mentor is a good match. This question, along with the possible answers, is provided in Figure 2.

Because each mentee is given the opportunity to choose the most important preference, we incorporate a weight on certain characteristics identified from their response

Table 1. A listing of the eight criteria identified from the existing survey. Each criteria has an associated matching matrix, whose elements represent a connection between mentees and mentors in that respective category.

	Matrix	**Criterion**	**Description**
Weighted by Question 9	Y_1	Age Preference	Age range in which the mentee prefers the mentor.
	Y_2	Academic Discipline	Academic disciplines of mentee and mentor are the same or similar.
	Y_3	Career Interests	Mentee's career interests align with mentor's current or future career.
	Y_4	Language Preference	Languages, other than English, in both mentee and mentor prefer.
	Y_5	Challenges Faced	Mentee and mentor have had similar life experiences.
Uniform	Z_1	Gender	Mentee and mentor have similar gender preferences.
	Z_2	First Generation	Alignment of first-generation college students.
	Z_3	Race/Ethnicity	Mentee and mentor have similar race/ethnicity profiles.

Q9. Out of the preferences you shared, which is most important to you?

1.) *My mentor is about the age I requested.*
2.) *My mentor and I have the same academic interests.*
3.) *My mentor and I share similar career interests.*
4.) *My mentor and I speak the same language (besides English).*
5.) *My mentor and I have faced similar challenges in life.*
6.) *All of these are equally important to me.*
7.) *Something else matters most to me.*

Figure 2. A snapshot of Question 9 of the existing survey that asked the mentee to identify the most important criteria.

to Question 9. From the eight notable linked survey questions, we construct five criteria that are assigned a weight and three minor criteria that have uniform weights applied. The minor criteria are weighted uniformly with a small numerical value. These criteria essentially act as a tie-breaker. Hence, our eight criteria for determining compatibility come in two flavors: criteria weighted by Question 9 and uniformly weighted criteria (see Table 1).

For each weighted criterion, we construct an $m \times n$ matrix Y_k, for $k \in \{1, 2, 3, 4, 5\}$. Each element of Y_k, denoted by y_{ijk}, is a numerical value between 0 and 1 that measures whether or not matches occurred in the corresponding linked survey questions. Likewise, we construct $m \times n$ matrices Z_ℓ, for $\ell \in \{1, 2, 3\}$, that correspond to each uniform criterion. The following descriptions demonstrate our methodology for constructing the numerical values y_{ijk} and $z_{ij\ell}$ in each set of matrices:

- **Criteria weighted according to Question 9:**
  - Y_1: Age Relation – A connection between s_i and t_j exists if the age of t_j is greater than the age of s_i by 5 or more years. That is,

$$y_{ij1} = \begin{cases} 1 & \text{if (age of } t_j) - (\text{age of } s_i) \geq 5 \\ 0 & \text{otherwise} \end{cases}$$

    Data for this particular criterion were limited. Here, we simply assume that the mentee prefers someone either near to their age or older. A more dynamic approach is applied in the second iteration of the model.

  - Y_2: Academic Discipline – The drop-down answer list for this criterion has 108 options, where the participant only selects one. A connection between s_i and t_j exists if the mentee's discipline is relatively "close" to the mentor's discipline. We define this distance from one discipline to another by constructing a graph (i.e. tree) by inspection of all 108 disciplines. A depiction of this tree is given in Figure 3. Two different disciplines are connected with an edge if they are similar in nature. An example of directly related studies are mathematics and physics. These two disciplines would be connected by a single edge. Indirectly related subjects are mathematics and mechanical engineering since mechanical engineering can be viewed as directly related to physics. Given a s_i's academic discipline, a connection between s_i and t_j is determined by traversing this tree. A similarity score is created by the

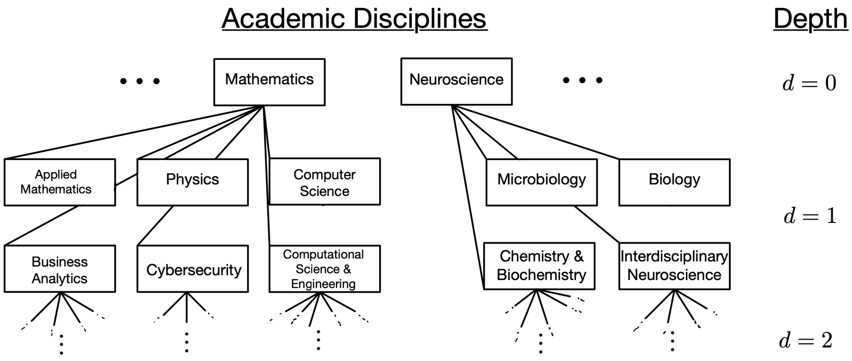


Figure 3. A snapshot of the related academic disciplines represented by a graph.

following function:

$$y_{ij2} = \begin{cases} b^d & \text{if } 0 \leq d \leq 2 \\ 0 & \text{if } d > 2 \end{cases},$$

where b is the branching factor and d is the depth. To find d, a breadth-first search is done on a graph connecting all of the majors. However, since breadth-first search is an expensive algorithm ($O(b^d)$), we limit the depth to 2 to ensure compatibility between the mentor and mentee's fields of study. For the purpose of this study, we use $b = 1/3$. For the exact definition of which majors are directly related, see the supplementary information.

- Y_3: Career Interests – The drop-down answer list for this criterion has 26 options with the possibility to list more than one. A connection between s_i and t_j exists if at least one match between all provided answers to the corresponding survey questions is made. The numerical value is calculated as the number of matches divided by the total number of mentee responses to the question. For instance, if s_i lists career interests as

  {Arts and culture, Communications, Community and social services}

  and t_j lists career interests as

  {Arts and culture, Education, Education administration, Government},

  then $y_{ij3} = 1/3$. This is done to avoid penalizing those students who list very few interests, as opposed to those who list many interests. In this way, the weights are biased in the mentees favor.

- Y_4: Language Preference – The drop-down answer list for this criterion has 16 options with the possibility to list more than one. A connection between s_i and t_j exists if at least one match between all provided answers is made, *not including* English. Question 4 from each survey (see Table 1) is worded in such a way that Grad LEAP seeks a connection between preferred language, which may not be English. Furthermore, answer 2 from Question 9 excludes English. Because every mentee and mentor is taking or has taken English as a Second Language (ESL) courses at the University of Delaware, we effectively take English out of the calculation for y_{ij4}. That is, we assume every participant will match for English and only compute a value for y_{ij4} based on other provided languages in the response. The numerical value is calculated as the number of non-English matches divided by the total number of mentee non-English responses to the question. To demonstrate, suppose s_i selects

$$\{\text{English, Mandarin, Vietnamese}\},$$

and t_j selects

$$\{\text{English, Mandarin, Cantonese}\},$$

then $y_{ij4} = 1/2$. This is done to avoid penalizing those mentees who list very few interests, as opposed to those who list many interests.

  - Y_5: Challenges Faced – The drop-down answer list for this criterion has 22 options with the possibility to list more than one. A connection between s_i and t_j exists if at least one match between all provided answers to the corresponding survey questions is made. Similarly to the previous two criteria, the numerical value, y_{ij5}, is calculated as the number of matches divided by the total number of mentee responses to the question.

- **Uniformly weighted criteria:**
  - Z_1: Gender – A connection between s_i and t_j exists if the gender selected in the drop-down menu matches exactly. That is,

$$z_{ij1} = \begin{cases} 1 & \text{if (gender of } s_i) = (\text{gender of } t_j) \\ 0 & \text{otherwise} \end{cases}$$

  - Z_2: First Generation – Indication of whether or not a participant was a first generation college student was incorporated into a much broader survey question on background. However, we were provided only with "TRUE" or "FALSE" data about the first generation component of that survey question. Therefore, a connection between s_i and t_j exists if the true/false selection matched exactly. That is,

$$z_{ij2} = \begin{cases} 1 & \text{if both true or both false} \\ 0 & \text{otherwise} \end{cases}$$

    If a participant left this entry blank, no match was made on first generation status.
  - Z_3: Race/Ethnicity – The drop-down answer list for this criterion has 12 options with the possibility to list more than one. A connection between s_i and t_j exists if at least one match between all provided answers to the corresponding survey questions is made. The numerical value, z_{ij3}, is cal-

culated as the number of matches divided by the total number of mentee responses to the question.

Now that the criteria are defined, we associate a weight to the weighted criteria, Y_k, based on the mentee's selection to Question 9. We note from the possible selections to this question, that a mentee could respond with "All of these [criteria] are equally important." They could also respond with "Something else matters most to me" that is not included in the criteria. Otherwise, the other options are to select a single weighted criterion that is the most important. Therefore, we construct the matrix V_k whose elements v_{ijk} are the associated weights to the criteria element y_{ijk}. Responses 1 through 5 correspond directly to criteria $k = 1$ through $k = 5$. Suppose a is the participant's response to Question 9, so that $a \in \{1, 2, \ldots, 7\}$. Let $\boldsymbol{J}$ represent the $m \times n$ matrix of ones, then we define V_k as a piecewise matrix function dependent on the participant's response:

$$V_k(a) = \begin{cases} .6\boldsymbol{J} & \text{if } a \in \{1,2,3,4,5\} \text{ and } a = k \\ .1\boldsymbol{J} & \text{if } a \in \{1,2,3,4,5\} \text{ and } a \neq k \\ .2\boldsymbol{J} & \text{if } a \in \{6,7\} \end{cases} \tag{1}$$

To demonstrate this, assume that the mentee selects answer 2, then the corresponding weight matrix for this participant is

$$V_k(2) = \begin{cases} 0.6\boldsymbol{J} & \text{if } k = 2 \\ 0.1\boldsymbol{J} & \text{if } k \in \{1,3,4,5\} \end{cases}$$

In essence, the weight of 0.6 is shifted to the appropriate criterion based on the mentee's response. We choose weights so that the sum over the criteria (k) is one. If the mentee chooses answer 6 or 7, we have

$$V_k(6) = V_k(7) = 0.2\boldsymbol{J}.$$

This yields equal weight for all criteria even in the case when the mentee selects answer 7. An equal weight distribution is justified in this case since we do not know in particular which criterion needs to be more heavily weighted.

For the uniform criteria, we use a uniform weight of $w = 0.01$. Here, the three uniform criteria are not mentioned in Question 9, and we use a much smaller weight so that the other, more important, criteria stand out. Essentially, the uniform criteria give the compatibility a slight "nudge" and may act as a tie-breaker.

2.2. Compatibility Score

Let C be the compatibility matrix of size $m \times n$ whose elements are the compatibility score, c_{ij}, of mentee s_i and mentor t_j, defined by

$$c_{ij} = \sum_{k=1}^{p} v_{ijk} y_{ijk} + \sum_{\ell=1}^{q} w z_{ij\ell}.$$

Here, $p = 5$ is the number of weighted criteria and $q = 3$ is the number of uniform criteria. The indices k and ℓ are as defined in the previous subsection, and y_{ijk} and

$z_{ij\ell}$ are criteria matrix elements. The value v_{ijk} (w) is the value of weighted (uniform) criterion k (ℓ) to mentee i. Here, the maximum value of y_{ijk} for all i, j, and k is 1. Further, the values of v_{ijk} for $k = 1, 2, \ldots, p$ are normalized so that $\sum_{k=1}^{p} v_{ijk} = 1$ for all i, j. This means, for any given i and j, $\sum_{k=1}^{p} v_{ijk} y_{ijk} \leq 1$. However, given any i and j, since $w = 0.01$, the sum $\sum_{\ell=1}^{q} w z_{ij\ell}$ is at most 0.03. Therefore, we have

$$0 \leq c_{ij} \leq 1.03, \qquad \text{for all } i \text{ and } j.$$

A compatibility score of 0 means the mentee and mentor disagree on all criteria. A score of 1.03 indicates a perfect match, including the slight "nudge" by the uniform criteria.

2.3. Assignment Problem Formulation

As previously mentioned, the goal of the algorithm was to find a set of good mentor-mentee pairings given a set of mentees and a set of potential mentors. The first method we utilize to compute a set of best possible matches through linear programming.

By generating the matrix C of compatibility scores for every possible mentee-mentor pair, we formulated the original goal into a classic form of a linear program known as the "Assignment Problem" [2]. In the assignment problem, there is a binary decisions variable x_{ij} for each possible mentor-mentee pairing. In the final solution of the linear program, x_{ij} takes on a value of 0 or 1 according to the assignment below:

$$x_{ij} = \begin{cases} 1 & \text{if } s_i \text{ is matched with } t_j \\ 0 & \text{otherwise} \end{cases}$$

The objective of the linear program is to maximize the objective function z which is defined below:

$$z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}.$$

In this form of the objective function, the total compatibility among all matches assigned is maximized. When maximizing the objective function z, the solution is constrained by two phenomena: (1) each mentee must be assigned exactly 1 mentor, and (2) each mentor must be assigned *at most* L_j mentees, where L_j is the number of mentees mentor j is willing to mentor. These constraints can be easily incorporated into the linear programming formulation, which is shown below:

$$\text{Maximize} \quad z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$

Subject to:

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad \forall\, i = 1, 2, \ldots, m$$

$$\sum_{i=1}^{m} x_{ij} \leq L_j \qquad \forall\, j = 1, 2, \ldots, n$$

$$x_{ij} \in \{0, 1\}$$

The assignment problem can be solved more efficiently than standard linear programming problems. Using the Hungarian method [2], the assignment problem can be solved in $O(n^3)$ time as opposed to $O(2^n)$ time, as would be the case using the Simplex Method.

However, the Hungarian method requires that each constraint be an equality constraint with right-hand side equal to 1. Also, we must have $m = n$. Our LP formulation can be manipulated to fit this structure by adding *"dummy"* mentors and/or *"dummy"* mentees, before solving. The dummy variables do not represent any real phenomenon, and are included simply for the purposes of initializing the Hungarian method.

To demonstrate the conversion to an assignment problem, consider the total number of possible mentor slots available, given by $n' = \sum_{j=1}^{n} L_j$. In order for every mentee to receive a mentor, we must have $n' \geq m$. Let $N = \max\{m, n'\}$. If there are more mentor slots than mentees ($n' > m$, $N = n'$), exactly $N - m$ dummy mentees are added with compatibility score $c_{ij} = 0$ with all mentors. Similarly, if there are more mentees than available mentor slots ($n' < m$, $N = m$), exactly $N - n'$ dummy mentors will be added before solving. Any mentor or mentee assigned to a dummy mentee or mentor in the optimal solution is simply unmatched in the final solution. In the case when $n' < m$, a suboptimal solution is obtained since exactly $N - m$ unmatched mentees exist.

Let $\hat{c}_{ij}$ and $\hat{x}_{ij}$ denote the extended compatibility score and extended binary decisions variable, respectively, which includes L_j duplicate mentors for each t_j and appropriate dummy mentees/mentors depending on the value of N. Additionally, since the right-hand-side of all the constraints is in integer form, properties of linear programming dictate that all $\hat{x}_{ij}$ in the final solution will also be integer valued. Thus, the integer constraint on $\hat{x}_{ij}$ can also be relaxed. The assignment problem then takes the form:

$$\text{Maximize} \quad z = \sum_{i=1}^{N} \sum_{j=1}^{N} \hat{c}_{ij} \hat{x}_{ij}$$

Subject to:

$$\sum_{j=1}^{N} \hat{x}_{ij} = 1 \qquad \forall\, i = 1, 2, \ldots, N$$

$$\sum_{i=1}^{N} \hat{x}_{ij} = 1 \qquad \forall\, j = 1, 2, \ldots, N$$

$$0 \leq \hat{x}_{ij} \leq 1$$

The final algorithm that was generated during this project was designed to be robust to unequal numbers of mentors and mentees, and omits the dummy assignments from the final output.

2.4. Stable Marriage Problem Formulation

The second method we utilize in our algorithm to find a good set of mentor/mentee matches is the stable-marriage problem (SMP), which is solved using the Gale-Shapley algorithm [1]. Having obtained the compatibility matrix C corresponding to the mentees $s_1, \ldots, s_m$ and mentors $t_1, \ldots, t_n$, the Gale-Shapley algorithm provides a solution to our matching problem that finds the *stable pairings* that are *mentee-optimal*. Stable matches are those such that there are no mentor-mentee pairs, say (s, t), such that both s and t would prefer to be matched together instead of in their current pair. The Gale-Shapley algorithm provides *stable* matchings, which may result in a different set of matchings than the solution found using the LP. Further, it can provide stable matchings which are optimal with respect to mentee preferences of mentors, meaning that that every mentee is as happy as they can be in a stable matching [1].

The algorithm can be briefly summarized as follows: Extend the set of mentees and mentors to include L_j duplicate mentors for each t_j and add appropriate dummy mentee/mentor variables to balance cardinalities of each set to N. Using the same compatibility matrix C (with zero entries for dummy variables), the Gale-Shapley algorithm constructs two families of preference orderings: an ordering $\leq_s$ of the mentors' $t_1, \ldots, t_N$ compatibility with each student s, and an ordering $\leq_t$ of the mentees $s_1, \ldots, s_N$ for each mentor t. Then, at each iteration, it takes an unmatched mentee and matches them to their top preference of mentor who will "accept" them. A mentor will accept a mentee if they are not already matched to a mentee they prefer; if the mentor is already matched but would rather switch, they do so, and we rematch the now unmatched mentee later. The process continues until all mentees are matched to the top mentors who will accept them.

Similarly to the LP formulation of the matching problem, the Gale-Shapley stable marriage algorithm is robust to cases where the number of mentors and mentees is unequal. Additionally, it is also possible to perform this algorithm with the roles of the mentees and mentors swapped, meaning the pairings will be *mentor-optimal*. In addition to finding the most optimal matching for mentees, the Gale-Shapley algorithm is fast, finding an optimal set of mentor-mentee pairings in $O(n^2)$ time.

2.5. Results of LP and SMP Models

We expect to see different results from the Hungarian method and the Gale-Shapley algorithm since they have different functionalities. Figure 4 demonstrates how well both algorithms can match mentor and mentee pairs[1]. The top subfigure is a cumulative frequency plot of the compatibility scores and the middle subfigure is a frequency plot of the same data. If the definition of a good pairing is based solely on the compatibility score, conclusions can be made using that data alone. SMP produces more pairs with very high and very low scores, while LP produces less of these extremes. This metric suggests SMP may be superior for a good pairing threshold of $c_{ij} \geq 0.4$, while LP is superior otherwise. However, the lower subfigure displays a different metric: the percent of matches that satisfy each individual weighted criterion. From this, LP appears to be the better choice if one cares solely about one of the criterion since it matches at least as many or more individual values than the SMP formulation.

Table 2 demonstrates more comparison metrics for the two algorithms using cur-

[1]The Gale-Shapley algorithm is dependent on the processing order of the mentors, so it will not always find the same solution. However, the algorithm is not very susceptible to initial conditions, so the relationship between the algorithms is consistent.

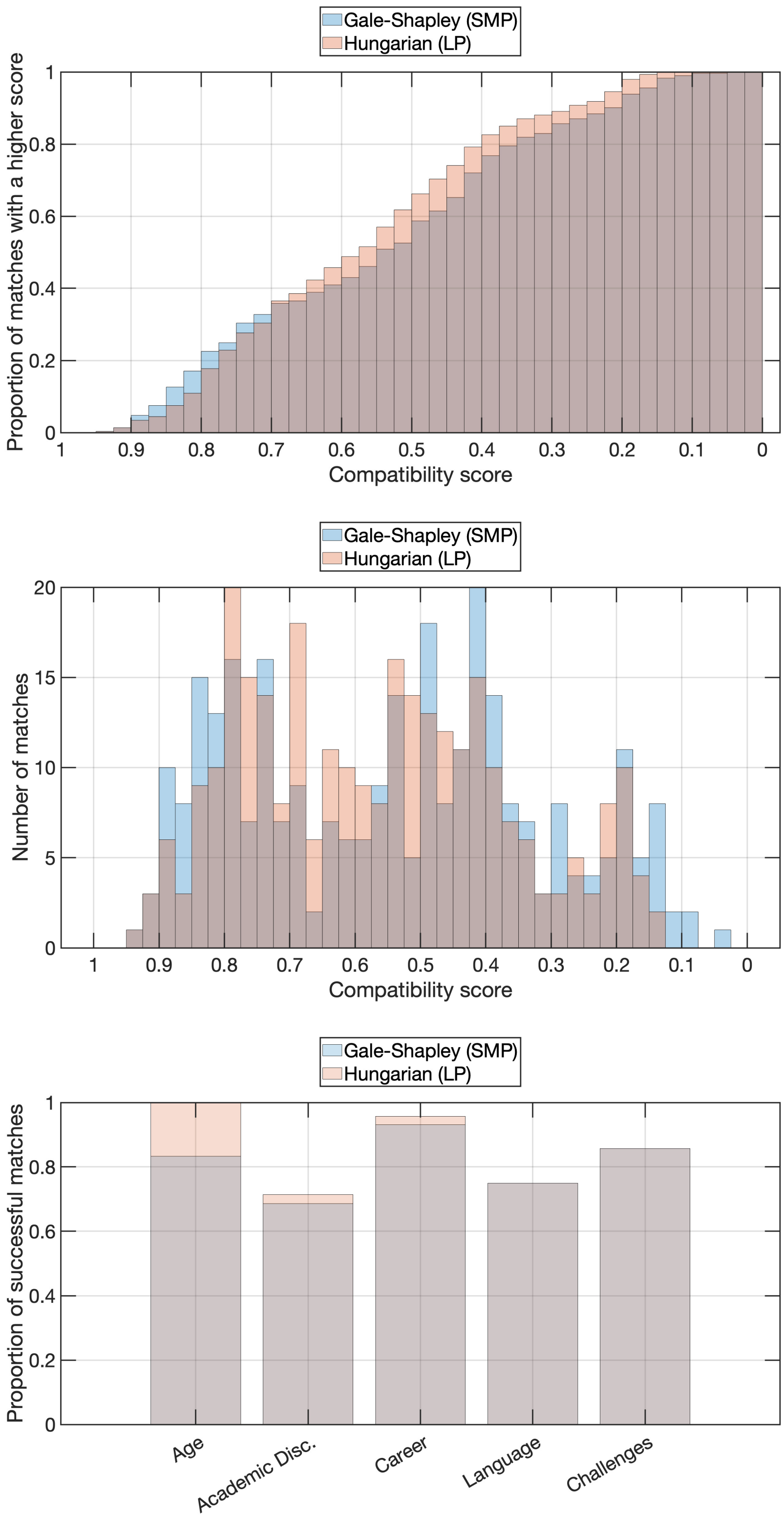


Figure 4. Comparisons of the Hungarian and Gale-Shapley solutions for the Cohort 3 data. Cumulative distribution (top) and frequency distribution (middle) of compatibility scores matched by each algorithm. (Bottom) Bar graph showing the proportion of matches that accurately pair each weighted criterion.

rent real data. The first row contains the percent of matches where the metric most important to the mentee was the same for the mentee-mentor pair. The second row shows what we denote the c-ratio, which is defined as follows

$$c\text{-Ratio} = \frac{z_{actual}}{z_{ideal}},$$

where z_{actual} is the z value obtained from the results of the algorithms and z_{ideal} is the z value if every mentee was paired with the mentor with whom they share the most similarity. The Gale-Shapley algorithm does not optimize this value, so it is unsurprising that the Hungarian method outperforms it. However, it still performs very well on this metric. The third row displays the percent of matches where *either* the academic disciplines of the pairs *or* the career of the mentor and career aspirations of the mentee exactly match. The fourth row shows the percentage of pairs where *both* academic disciplines *and* career interests matched. These metrics are considered heavily by the University of Delaware when matches are approved, so this metric is especially good at predicting the number of pairings that are successful. Hungarian outperforms the Gale-Shapley algorithm in all four of these metrics for the Cohort 3 dataset, but not to the extent that one can conclude that Hungarian is always the best algorithm to use. As a result, when building an executable file for the University of Delaware to use, the option of using either algorithm is included.

Table 2. Successful pairing metrics for each algorithm.

Metric	**Hungarian (LP)**	**Gale-Shapley (SMP)**
Top Preference	86.47%	83.57%
c-Ratio	86.75%	83.40%
Acad or Career	90.10%	86.69%
Acad and Career	35.49%	34.47%
Min c-Score	0.1304	0.0302
Mean c-Score	0.5636	0.5419
Max c-Score	0.9374	0.9499

3. Formulations Based on Revised Survey

We have successfully built two working models that match mentees to mentors with a relatively high success rate based on compatibility. Compatibility is determined by analyzing 8 questions from the current survey. Certain compatibility measures are weighted according to participants' answers to Question 9 (see Figure 2), which effectively allows users to determine the most important criteria for matching to a potential mentor. The models are validated using the Cohort 3 data set from this existing survey.

We discussed with our company liaison how the survey could be revised in order to refine the compatibility score while simultaneously making the program structure more robust. In the subsections that follow, we discuss the changes made to the survey, and how these changes affect the existing criteria and create new criteria. We end this section with a discussion of the results of both models built on synthetic data from our new survey.

3.1. *Revised Survey*

We made several modifications to the existing survey. We removed unnecessary questions (e.g. a cultural question seemed redundant with the background, race/ethnicity, and language questions), added a question about mentee-mentor degree preference, and added a Likert importance scale for 10 different preferences. Unlike the previous survey question that only allowed students to identify one important preference, the Likert scale method gives them the freedom to decide the importance of all possible match criteria.

The revised survey now has all of the mentees rank the importance of each question on a scale from 0 to 5. Figure 5 shows an example of a new survey question. By doing this, the values of v_{ijk} (for all j) can be personalized to each mentee in a more complex way than the original formulation. In the previous formulation, v_{ijk} was calculated such that v_{ijk} is large for the criterion k that student i selected as most important to them, while for all other criteria the value of v_{ijk} is smaller and uniform (see Equation (1)). This scheme explains why the results on the original data show a vast difference between the percentage of mentees who matched either close academics or one exact career match ($\approx 90\%$) and both close academics and one exact career match ($\approx 35\%$) since most mentees choose one of these criteria as their most important, which is weighted very heavily. However, past feedback suggests that career interests and academic metrics are both very important to mentees. By incorporating this scale, the overall satisfaction of each individual mentee can be increased.

Other revised questions include a variant to the age preference where mentees select

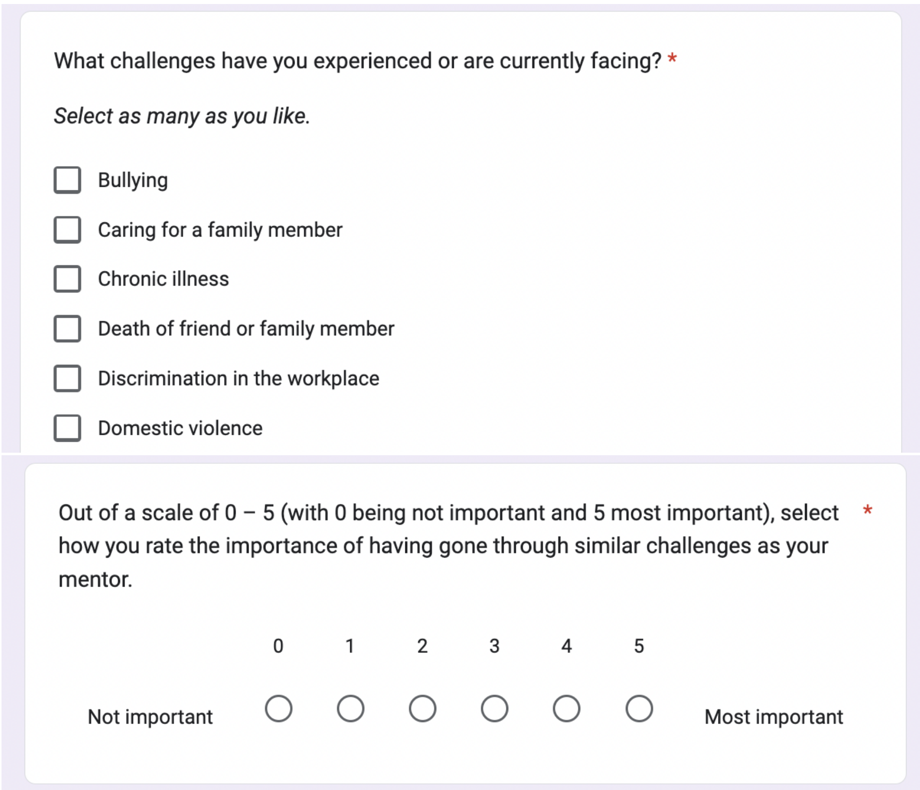
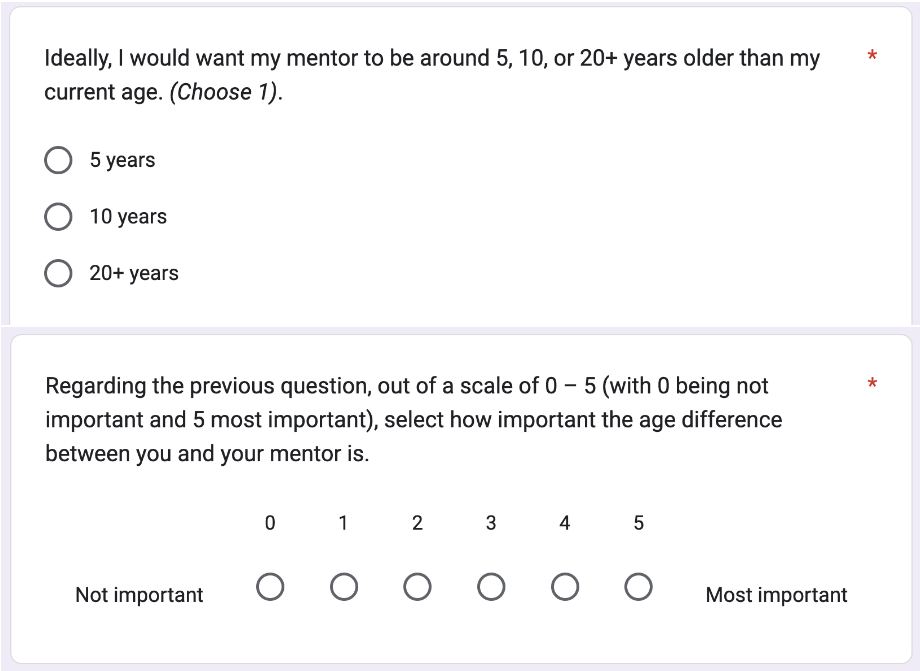


Figure 5. An example question about life experiences/challenges faced on the mentee survey that includes the new Likert scale.



Ideally, I would want my mentor to be around 5, 10, or 20+ years older than my current age. *(Choose 1)*. *

- 5 years
- 10 years
- 20+ years

Regarding the previous question, out of a scale of 0 – 5 (with 0 being not important and 5 most important), select how important the age difference between you and your mentor is. *

	0	1	2	3	4	5	
Not important	○	○	○	○	○	○	Most important

Figure 6. The modified question on age preference with the accompanying Likert scale. This question is used to better gauge the mentee's preferred mentor age.

whether they would prefer to have mentors around 5, 10, or over 20 years older than them. This provides a concrete metric to determine how the age compatibility between the mentor and mentee can be generated without complicating the system to a large degree. This question is shown in Figure 6. Another is a variant of the language question where instead of mentees and mentors listing all of the languages they feel comfortable speaking, they only select the languages other than English that fit this category. Since the mentors and mentees both have or are attending the University of Delaware, it is known that they can speak English. As a result, whether participants would prefer or are willing to speak other languages is more pertinent information. Finally, another question that only exists on the mentor survey is gender preference. The answer to this question acts as an "override" feature, where if the mentor is not willing to work with a mentee of a specific gender identity, then the compatibility score will be set to zero. These changes were implemented into a new survey and the associated Google Forms were created. In the following subsections, we explain how the new compatibility score is created from the altered criteria.

3.2. New and Modified Criteria

Using the updated survey format, we modify criteria that we defined in Section 2.1 and add new criteria that aligns with new survey questions. Unlike our previous model, not only will the importance rank from the Likert scale influence the weight of the criteria, it may also affect the strength of a potential match.

- **Modified Criteria:**
  - Y_1: Age Relation – We expand upon our previous formulation by including a wider range of possible compatibility scores based on age match. A

connection between s_i and t_j is determined using a custom, piecewise function. Consider two input variables: mentee i's age, a_i, and mentor j's age, b_j. Also, consider two parameters: the mentee's preferred age range of the mentor, β, and the importance rank, γ. The two parameters are given by the mentee answering the question in Figure 6. Here, if the mentee selects "around 5 years older," the value of β is 5. If the mentee selects "20+ years older," the value of β is 20. Therefore, $\beta \in \{5, 10, 20\}$. For the ranking value, we have $\gamma \in \{0, 1, 2, 3, 4, 5\}$. We assume a connection between s_i and t_j is a perfect match if the mentor's age is within 5 years of the mentee's preferred range. That is, if $|b_j - (a_i + \beta)| \leq 5$, then $y_{ij1} = 1$. Outside this range, we define semi-compatibility by using a smooth function based on the logistic curve. We define the following functions:

$$f_{\pm}(a_i, b_j; \beta, \gamma) = \frac{2}{\gamma} \cdot \frac{1}{1 + e^{\pm 0.2[b_j - (a_i + \beta) - 5]}}$$

Note, the functions exist for $\gamma \neq 0$. Here, f_+ (f_-) is monotonically decreasing (increasing), and transitions smoothly from a maximum value of $2/\gamma$ to 0 (minimum value of 0 to $2/\gamma$). Furthermore, the inflection point of the logistic curve is at age $a_i + \beta + 5$, which is an arbitrary addition of 5 years to the mentee's preferred age range. The transition is biased in the "older" direction because of the nature of the question. These functions ensure that the matching value associated with a mentor's age will gradually taper off as the match is less satisfactory on either side of $y_{ij1} = 1$. We break down the full function below:

- If the mentee chooses 5 or 10 years older as the age preference, then the matching strength, y_{ij1}, takes on the following values dependent on the rank, γ:

$$\text{for } \gamma = 0, \quad y_{ij1} = 1;$$

$$\text{for } \gamma \in \{1,2,3,4\}, \quad y_{ij1} = \begin{cases} f_-(a_i, b_j; \beta, \gamma) & \text{if } (a_i + \beta) - b_j < 5 \\ 1 & \text{if } |b_j - (a_i + \beta)| \leq 5 \\ f_+(a_i, b_j; \beta, \gamma) & \text{if } b_j - (a_i + \beta) > 5 \end{cases};$$

$$\text{for } \gamma = 5, \quad y_{ij1} = \begin{cases} 0 & \text{if } (a_i + \beta) - b_j < 5 \\ 1 & \text{if } |b_j - (a_i + \beta)| \leq 5 \\ 0 & \text{if } b_j - (a_i + \beta) > 5 \end{cases}.$$

- If the mentee chooses 20+ years older (i.e. $\beta = 20$), then the matching strength, y_{ij1}, takes on the following values dependent on the rank, γ:

$$\text{for } \gamma = 0, \quad y_{ij1} = 1;$$

$$\text{for } \gamma \in \{1,2,3,4\}, \quad y_{ij1} = \begin{cases} f_-(a_i, b_j; \beta, \gamma) & \text{if } (a_i + \beta) - b_j < 5 \\ 1 & \text{if } (a_i + \beta) - b_j \geq 5 \end{cases};$$

$$\text{for } \gamma = 5, \quad y_{ij1} = \begin{cases} 0 & \text{if } (a_i + \beta) - b_j < 5 \\ 1 & \text{if } (a_i + \beta) - b_j \geq 5 \end{cases}.$$

A plot of these functions is shown in Figure 7. Here, we can see that if $\gamma = 0$, then $y_{ij1} = 1$ no matter what else occurs. In this case, if age preference is of no importance to the mentee, then any age is a perfect match. Lastly, in the case when the mentee selects 20+ years, the right-hand side does not taper off.

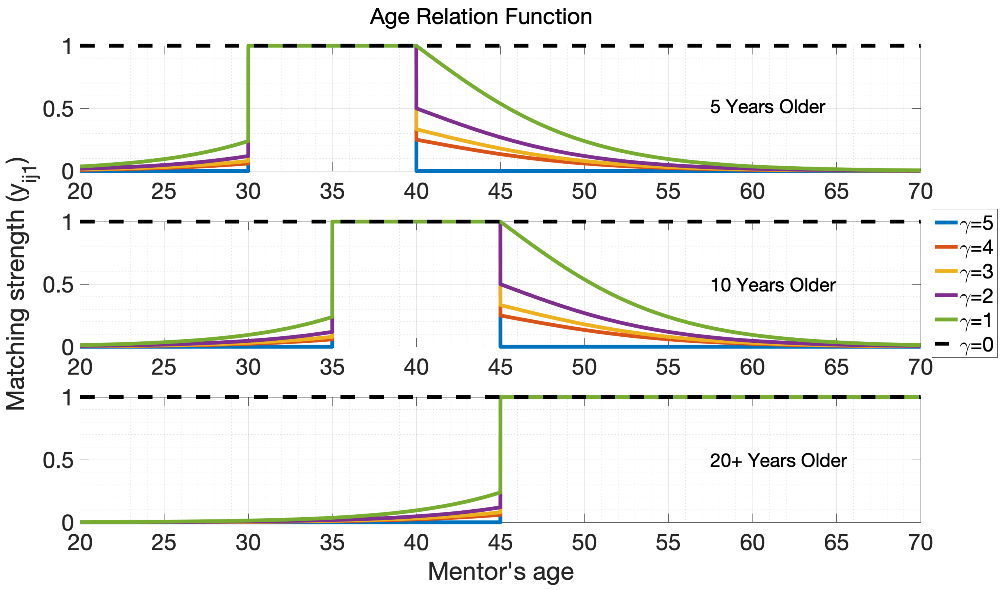


Figure 7. The piecewise function used for age preference separated by the mentee's selection to the question in Figure 6. To illustrate the function, we assume $a_i = 30$. The x-axis is b_j, which is mentor j's age. Mentee selects 5 years older (top), 10 years older (middle), or 20+ years older (bottom).

- Y_2: Academic Discipline – The value for y_{ij2} is calculated in an identical manner as in the previous model, except for when the mentee places the utmost importance on academic discipline. That is, if the mentee chooses a 4 or 5 on the Likert scale for the importance of the academic discipline category, we assign $y_{ij2} = 1$ when the mentee and mentor have identical disciplines. To demonstrate this, we let $\gamma \in \{0, 1, 2, 3, 4, 5\}$ represent the selected importance ranking, then

$$\text{for } \gamma \in \{0,1,2,3\}, \quad y_{ij2} = \begin{cases} b^d & \text{if } 0 \le d \le 2 \\ 0 & \text{if } d > 2 \end{cases},$$

$$\text{for } \gamma \in \{4,5\}, \quad y_{ij2} = \begin{cases} 1 & \text{if } d = 0 \\ 0 & \text{if } d > 0 \end{cases},$$

where $b = 1/3$ and d represent the branching factor and depth, respectively, of the academic discipline graph in Figure 3.

- **New Criteria:**
  - Y_6: Background – The drop-down answer list for this criterion has 19 options with the possibility to list more than one. A connection between s_i and t_j exists if at least one match between all provided answers to the corresponding survey questions is made. The calculation of the matching

strength is identical to that of the previous model for calculating y_{ij3}, y_{ij4}, and y_{ij5}. The numerical value, y_{ij6}, is calculated as the number of matches divided by the total number of mentee responses to the question. This variable is a broadened version of Z_2, which was first-generation college student. The question involved here has more options to choose from and thus creates more avenues to make a mentee-mentor connection on background.

- Y_7: Mentee Goals – The drop-down answer list for this criterion has 8 choices with the possibility to list more than one. Some mentee options include "build professional connections," "expand support network," and "receive advice on international obligations or issues." The corresponding mentor survey asks if they would be able and willing to help with or provide advice in these same categories. A connection between s_i and t_j exists if at least one match between all provided answers to the corresponding survey questions is made. The matching strength, y_{ij7}, is calculated identically as the previous criterion.

- Y_8: Gender Preference – This criterion is a generalized version of Z_1 from the first model. A connection between s_i and t_j exists if the gender selected in the drop-down menu matches exactly. That is,

$$y_{ij8} = \begin{cases} 1 & \text{if (gender of } s_i) = (\text{gender of } t_j) \\ 0 & \text{otherwise} \end{cases}.$$

  This criterion has an "override" option dictated by the mentor's response. On both surveys we identify mentee and mentor gender; however, the mentor is permitted to answer another question asking if they prefer to work with or not work with a specific gender. Therefore, depending on how they answer this question, the matching value is reset to 0 or kept at 1 depending on the mentee's gender.

- Y_9: Race/Ethnicity – The drop-down answer list for this criterion has 12 choices with the possibility to list more than one. A connection between s_i and t_j exists if at least one match between all provided answers to the corresponding survey questions is made. The matching strength, y_{ij9}, is calculated identically as criterion Y_7.

- Y_{10}: Degree Preference – The drop-down answer list for this criterion has only 3 options, where the participant only selects a single answer. We calculate the matching value as follows:

$$y_{ij10} = \begin{cases} 1 & \text{if (degree preference of } s_i) = (\text{degree obtained by } t_j) \\ 0 & \text{otherwise} \end{cases}.$$

We now construct the weights for each of the newly defined criteria above. Since each criterion has a linked ranking, we redefine $k \in \{1, 2, \ldots, 10\}$ as the criteria index (i.e. $p = 10$). Based on the ranking that each mentee s_i submits, we let ω_{ik} be the weight that student i gave to criterion k. Therefore, $\omega_{ik} \in \{0, 1, 2, 3, 4, 5\}$ for all i and

k. Keeping the notation V_k to represent the $m \times n$ matrix of weights for each criterion k, we define a normalized weight element v_{ijk} using the following formula:

$$v_{ijk} = \frac{\omega_{ik}}{\sum_{k=1}^{p} \omega_{ik}} \qquad \forall j \in \{1, 2, \ldots, n\}.$$

In the rare case that mentee s_i selects $\omega_{ik} = 0$ for all k, we take this to mean that all criteria are weighted equally and set $v_{ijk} = 1/10$ for all j, k.

Using this definition for the weights, we now define the compatibility matrix C whose elements are

$$c_{ij} = \sum_{k=1}^{p} v_{ijk} y_{ijk}$$

Unlike the previous model, The normalization ensures that $0 \leq c_{ij} \leq 1$ for all i, j. That is, no "tie-breakers" or uniformly weighted variables exist. We now solve the two formulations based on this new definition of compatibility with simulated data.

3.3. Results of LP and SMP Formulations

The revised survey includes Likert scales for the mentee to rank each criterion that is important to them. This didn't exist on the original survey. At the time of this workshop, we had Cohort 3 data associated to the existing survey. Some of these data were transferred to the new survey by essentially moving existing answers. For example, the original data included all 293 mentee responses for academic discipline. We salvaged this data by moving it to spreadsheet based on the same question on the new survey. However, a significant portion of data does not exist, especially the rankings using the Likert scale. To validate our models using the new survey format, we synthetically generated data to fill these gaps. Likert scale Data that did not exist at all was uniformly randomly generated. The age range preference (5 years older, 10 years older, or 20+ years older) was also assigned uniformly. After analyzing the data for Cohort 3, we were able to get a similar distribution for several rankings based on mentees' answers to Question 9 (see Figure 2). For example, approximately 120 of the 293 respondents answered that academic discipline was the most important matching criteria. Therefore, we used a binomial distribution for this criterion with the appropriate parameters so that approximately 120 of the mentees' answers are a 4 or 5. All other write-in answers were moved from the Cohort 3 data. Therefore, we have a synthetically generated data set that is similar in nature to Cohort 3.

The results are shown in Table 3. The Acad or Career and Acad and Career metrics are notably much less than that of the real data, but that is due to the uniform distribution of rankings for unknown data such as age preference or gender. In reality, many mentees say that academic major and career interest are more important to them than factors such as age. However, the c-Ratios are improved to a great extent. As with the previous results, we see that Gale-Shapley generally finds that higher match but at the expense of having a lower minimum compared to that of the Hungarian algorithm. The range of c values is generally wider using the Gale-Shapley method while the Hungarian method attempts to stay closer to the mean and give every mentee a decent matching. Overall, the results are similar in terms of thresholding. The c-Ratio value is an important statistic here since mean z values of 0.5346 and

Table 3. Successful pairing metrics for each algorithm using revised survey with synthetically generated data.

Metric	Hungarian (LP)	Gale-Shapley (SMP)
c-Ratio	95.32%	93.50%
Acad or Career	73.38%	71.33%
Acad and Career	20.82%	23.55%
Min c-Score	0.3335	0.1319
Mean c-Score	0.5450	0.5346
Max c-Score	0.8152	0.8261

0.5450 are, on their own, not an ideal case. However, the large c-Ratio suggests that this may be true due to the entire pool of mentees and mentors not sharing as high compatibility as in the actual Cohort 3 data.

Figure 8 shows a cumulative distribution of compatibility scores as well as a frequency distribution using each algorithm. It is notable from the plots that the frequency of matched compatibility scores appears to be distributed more Normally than the prior. The v_{ijk} values are closer to uniformly distributed in this version of the assessment, so that may be the cause of this phenomenon. However, we see that the algorithms preform similarly with these data. The Gale-Shapley solution produces a few more matches with a high compatibility but the Hungarian method balances with a higher proportion of matches near the mean. From the frequency distribution, we see that the SMP formulation does produce a wider range of compatibility scores with one match having a very low score of 0.1319.

The results presented here are based on randomly generated data that replicates Cohort 3 to some extent. To get a better idea of how each model performs, we run a Monte Carlo simulation using 1000 hypothetical data sets. In each data set, the age preference and rankings are randomly distributed. A subset of the rankings follow a binomial distribution based on Cohort 3 data and other subset follows a uniform distribution.

The results of the Monte Carlo simulation are shown in Figure 9. We can see that both models have similar c-Ratio distributions with the LP model consistently doing slightly better for the majority of the trials. The Hungarian method has a mean c-Ratio of 0.9526, while the Gale-Shapley method's is 0.9313. The SMP formulation has slightly more variation. As in the prior model using Cohort 3 data, the LP formulation does better on this statistic since compatibility score is built into the objective function.

Referencing the three figures in the middle of Figure 9, the minimum matched compatibility score for Gale-Shapley is generally lower on average than the Hungarian method. This makes sense as the SMP formulation tends to get higher matches at the expense of some having low compatibility. The average minimum c-Score is 0.1897 for SMP, while the average minimum c-Score for LP is 0.2789. The difference in mean minimum values is notable and the SMP formulation tends to have a slightly higher variability for this statistic as well. The mean matched c-Score is relatively the same for each algorithm, with the Hungarian method edging out with a mean of 0.5449 to 0.5330. Variability is much lower for this statistic. The distributions of maximum c-Scores are also relatively equal, with the Gale-Shapley winning slightly with 0.8616 to 0.8597. This is reasonable since the Gale-Shapley finds highly compatible matches first. The variation in this statistic is essentially the same for each model. The LP formulation still achieves about the same amount of high c-Scores as the SMP model. Knowing the minimum c-Score is relatively low for SMP compared to LP, the trade-off

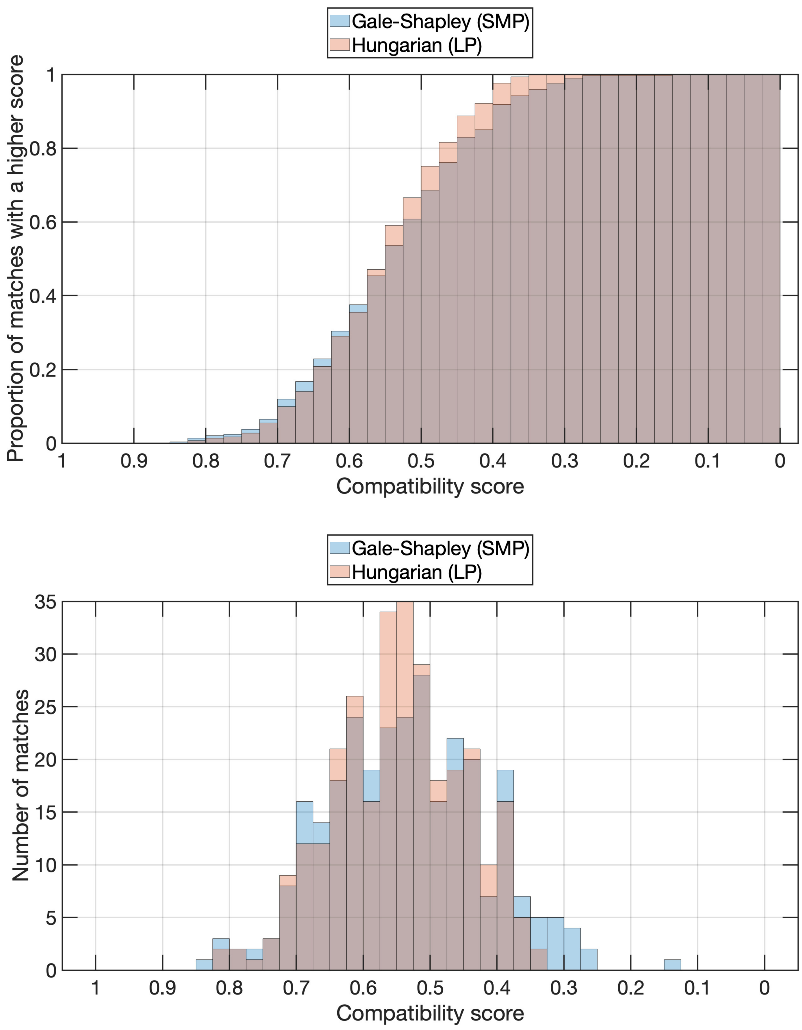


Figure 8. Comparisons of the Hungarian and Gale-Shapley solutions for a synthetic data set that preserves some characteristics of Cohort 3. Cumulative distribution (top) and frequency distribution (bottom) of compatibility scores matched by each algorithm.

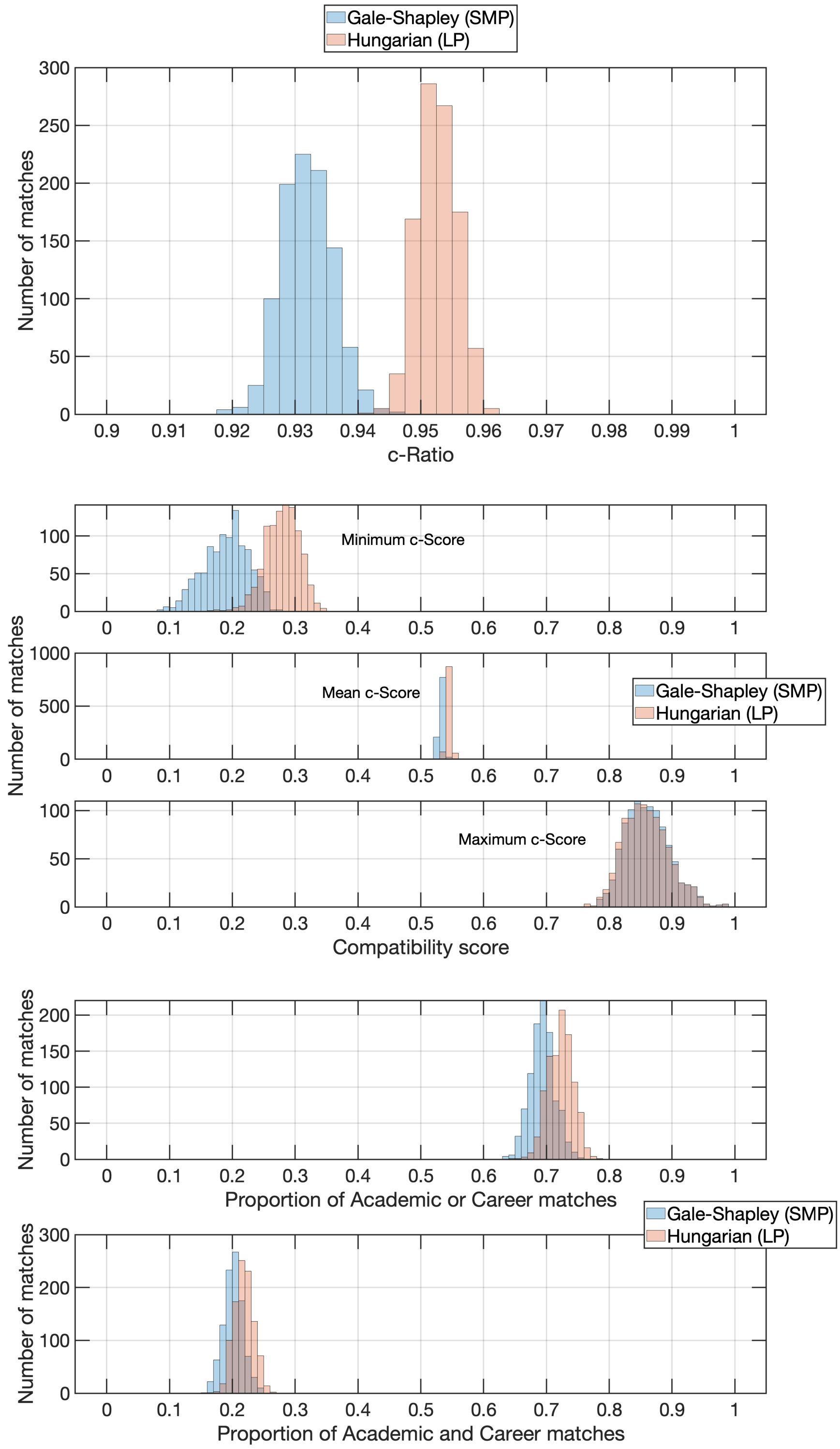


Figure 9. Results of the Monte Carlo simulation using 1000 hypothetical data sets. (Top) Frequency distribution of c-Ratios for each algorithm. (Middle) Frequency distributions for minimum c-Score, mean c-Score, and maximum c-Score. (Bottom) Frequency distributions for proportion of matches that successfully paired academic discipline *and/or* career interests.

between obtaining that higher c-Score at the expense of poor compatibility for a few other matches may not be worth it.

Finally, at the bottom of Figure 9, we see the performance of each algorithm on successfully pairing mentees and mentors with the same academic discipline and/or career interests. The average proportion for discipline *and* (*or*) career is 0.6947 (0.2015) for SMP and 0.7222 (0.2185) for LP. The means for each model are relatively the same with similar variation, but the Hungarian method is slightly better.

3.4. Deliverables

The two algorithmic solutions presented above can be used by the UD Graduate College for future cohorts of the Grad LEAP program. The work done on this problem resulted in two unique deliverables:

1. A new Google Forms survey that can be administered to potential mentees and mentors;
2. A user-friendly program (executable file) that takes the survey-generated Excel data as input, and outputs two Excel files of proposed mentor/mentee pairings, one generated using the assignment problem formulation, and one generated using the SMP formulation;

both of which are described in detail below.

As mentioned above, one of the main goals of this project was to develop a more robust, user-friendly algorithm that finds "good" mentor/mentee matches given data from a Google Form. During the development of the final algorithm, we created a new survey that asks similar questions to the original survey brought to the workshop by the University of Delaware Grad LEAP program. One of the main features of the new survey is that graduate students may now assign a value of importance to each individual question on a Likert scale, allowing them to create a unique set of weights for the question that allows the final algorithm to make the best match for the student's interests. An example question is provided below in Figures 5 and 6. In the final survey, there are 10 questions that ask the student about the following topics, as well as how important it is that their answers match with the answers of their mentor:

1.) Age gap preference	6.) Personal identities
2.) Academic discipline	7.) Goals of mentorship
3.) Career interests	8.) Gender identity
4.) Languages spoken	9.) Race/ethnicity
5.) Challenges faced in life	10.) Mentor academic degree attained

After students and mentors fill out the Google Form surveys, the results can be exported to Excel files (.xlsx), one with all the student data, and one with all the mentor data.

The second deliverable of this work is a user-friendly algorithm that can be used by the Grad LEAP program at the University of Delaware to generate a list of potential mentor/mentee pairings. To generate this deliverable, the final algorithm was coded in Python and was eventually converted to executable files that can be run on any kind of computer.

In the python code, Excel files of the mentee and mentor survey data sets are imported using *Pandas*. From there, the survey data is converted to numerical values using the methods described above, and the compatibility matrix C is calculated. Then, the code finds optimal mentor/mentee pairings using both the assignment problem formulation and the SMP formulation, and generates Excel files containing the pairings, along with their compatibility scores (c_{ij}). To help inform assignment decision-making, both the mentors' and mentees' career interests and academic interests are reported in the output file as well, along with the mentee's best possible mentor match.

In order to run on both Mac and Windows computers, the python code was converted into two executable files, a .exe file for Windows machines (called *Matching_Algorithm_Windows.exe*), and a Unix executable file for Macs (called *Matching_Algorithm_Mac*). *Matching_Algorithm_Mac* does not have a file extension, but will automatically be executed by Terminal on Mac machines. Both of these files can be run on and computer to execute the final algorithm, all that is needed is the Excel files with the mentor and mentee survey responses. Details on how to execute these files properly is included in an "Instructions" document along with the executable files and sample data files.

4. Machine Learning Approach

One approach that could be used is a classification model. To do this, a column of data for the mentee is compared to the mentor's data, and is given a score of 1 or 0. If the mentor and mentee are a good match based on the data being compared, they score a 1, otherwise they score a 0. Since there are only two outputs, we can use a classification model to determine if the mentor and mentee are a good match or not. Two different algorithms are used for determining the strength of a potential match: a decision tree and a random forest model. In the sections that follow, we construct each model and compare the results.

4.1. Decision Tree Model

The decision tree model is the first model that was implemented. The decision tree creates a split on a specific decision variable based on the Gini index, which is a measure of confidence for creating a node. A train/test split ratio of 40% was used to predict how effective the model would be in a real-life scenario. Figure 10 shows the distribution of accuracy scores resulting from the decision tree model for 1000 randomly generated training states. Further, Table 4 shows several scores resulting from the decision tree model by comparing academic discipline and career interests of each mentee/mentor.

Table 4. Scores for the decision tree model.

Metric	Academic Discipline	Career Interests
Accuracy Score	0.77	0.74
Precision Score	0.75	0.79
Recall Score	0.78	0.73
Cohen-Kappa Score	0.54	0.48

From the data, we could see that a matching based on majors alone was correct about 77% of the time. From the precision score and given a good match, the model is right about 75%. Using the recall score, it can correctly predict a good match about 78% of the time. From the Cohen-Kappa score we can interpret that the model will agree with manual pairing about 54% of the time. While these results may appear

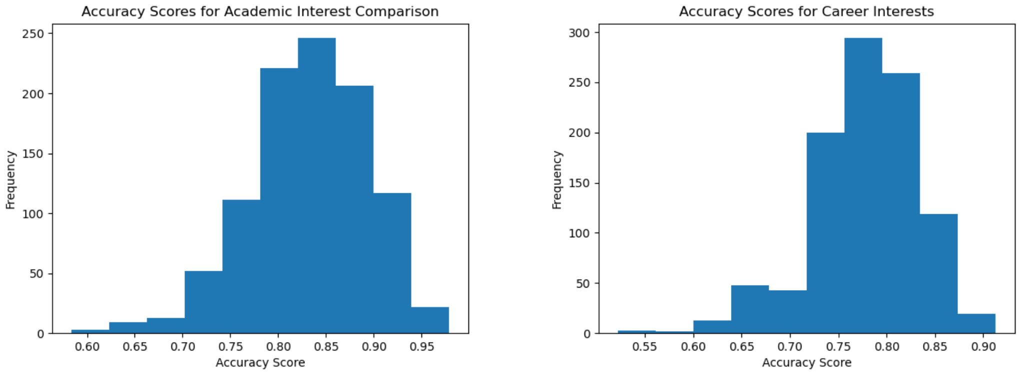


Figure 10. Accuracy scores of the decision tree model for academic discipline and career interests for 1000 random training states.

like a success, there are some flaws in the model. Firstly, a limited data size for the training data may have caused the model to not optimally create a decision tree for the data. This can be seen with the career interests, since there are more parameters that need to be compared than with academic discipline alone. Secondly, depending on the training data, a bias may occur within the tree.

In order to minimize bias from the training data, the goal is to minimize the standard deviation of the accuracy scores, as well as just observing the outliers to see if they make sense, in this case, we want to beware of maximums. While the minimums are bad objectively, by having a bad accuracy score, the maximums would need further testing with different training data in order to verify the accuracy. Below is a numerical analysis of the histogram data.

Table 5. Accuracy Score statistics for the decision tree model.

Statistic	Academic Discipline	Career Interests
Minimum	0.58	0.52
Maximum	0.98	0.91
Mean	0.84	0.77
Median	0.83	0.78
Standard Deviation	0.065	0.058

As seen in Table 5, the maximums of 0.98 and 0.91 are unrealistic and should be analyzed with further testing data. While the mean is high, with a correct prediction 84% of the time for majors and 77% of the time for career interests, it must be taken into account that the standard deviation is also large. With a large standard deviation, an assumption could be made that an average decision tree may bias possible outlier matches. Ideally, we would want a model that can handle the bias, and perhaps create cases for those outlier matches.

4.2. Random Forest Model

A random forest model is the second model that is implemented to solve the classification problem. A random forest works by using multiple decision trees to create

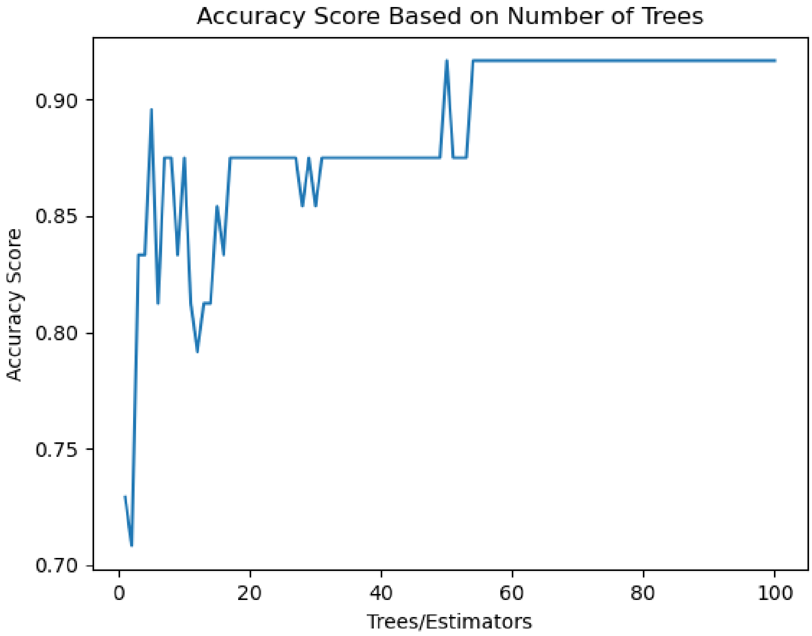


Figure 11. Optimizing the number of estimators using academic discipline as the matching criteria.

a prediction, unlike the single decision tree used in the first model. In order to keep consistency between the two models, a train/test split ratio of 40% was used again. One downside is that the amount of trees in the random forest, known as estimators, must be optimized beforehand. The graph in Figure 11 shows the accuracy score based on the amount of trees in the forest.

One observation that should be made from the graph is the convergence of the accuracy score as the number of estimators increase. While it is correct to say that if we make the amount of trees large that the model will be optimal, this lack of optimizing beforehand will cause an unnecessary increase in runtime, since there will be more trees to traverse. The optimal amount of trees for the major random forest is 50 trees, and the optimal amount of trees for career interests is 95 trees. Using those values, we present the results of the random forest accuracy scores for 1000 random training states in Figure 12. We can also calculate statistics about the random forest model. These statistics are shown in Table 6.

Table 6. Scores for the random forest model.

Metric	**Academic Discipline**	**Career Interests**
Accuracy Score	0.91	0.72
Precision Score	0.92	0.85
Recall Score	0.92	0.63
Cohen-Kappa Score	0.83	0.45

As we can see from the data, the random forest for major comparison has a Cohen-Kappa Score of 0.83, which is a drastic improvement over the decision tree model. For career interests however, there wasn't a large change in the scores. With all that being

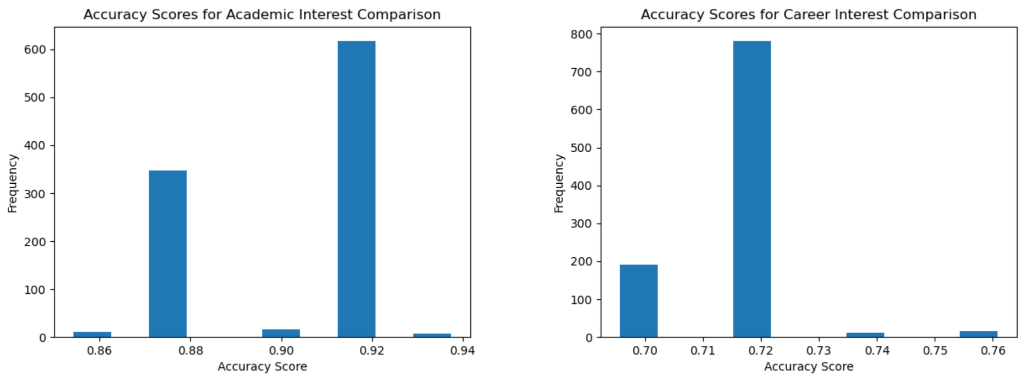


Figure 12. Accuracy scores of the forest model for academic discipline and career interests for 1000 random training states.

said, the precision score for career interests with the random forest is a increase from the decision tree model, with the trade off that the random forest has a worse recall score than the decision tree. Overall, the random forest is still better at identifying a good match for career interests, it just is not better at identifying a bad match.

One of the biggest problems with the decision tree was the underlying bias from the selection of the training data, and that to eliminate bias, the standard deviation of the accuracy scores should be minimized. With the random forest, our goal was to minimize the bias from the decision tree. These results can be seen in Table 7.

Table 7. Accuracy score statistics for the random forest model.

Statistic	**Academic Discipline**	**Career Interests**
Minimum	0.83	0.69
Maximum	0.94	0.76
Mean	0.90	0.71
Median	0.92	0.72
Standard Deviation	0.022	0.011

The goal of minimizing the standard deviation was a success. This means that the new model should answer different training data more accurately if presented with it. Another observation is the discontinuity in the bars of the histograms. This may have been caused from yet another bias in the training data. However, the difference in accuracy score bars is much smaller than those in the decision tree model.

4.3. Future Implementation

With the models above, a successful classification model was posed for each question in the survey. However, while this model is usable, there exists a flaw in the classification model: the model doesn't score a pairing, but instead just states if the match is good or bad. In order to get a score for matches, we need a regression algorithm, but there is no test data for a regression algorithm, since the given data doesn't have scores associated to the matches. To do this, we use the classification models to give us a score. Let $\vec{m}$ be the vector of matched data types, and let $\vec{s}$ be the importance scores of each question for mentees. The score for a matching would be as follows:

$$\text{score} = \frac{\vec{m} \cdot \vec{s}}{\sum_{k=1}^{p} s_k}$$

where p is the number of matching criteria. The proposed scoring formula has a range between 0 and 1, where if the mentor and mentee are a perfect match, $\vec{m}$ would be a vector of 1's, so the numerator and denominator would be equal. Applying this scoring formula to a large dataset would allow a regression model to be successfully created.

5. Conclusions

The objective of this workshop was to develop a practical model for determining optimal mentor-mentee pairings for the UD Graduate College's GradLEAP mentorship program. We believe this objective has been successfully met through the use of two distinct mathematical approaches: a linear programming formulation, which can be reinterpreted as an assignment problem and solved using the Hungarian method, and a stable marriage problem formulation, solvable via the Gale-Shapley algorithm. Both methods are integrated into a user-friendly executable file (.exe). The input for this program is an Excel sheet containing all mentor and mentee responses from a restructured Google Form survey, and the output is a list of optimal pairings generated by each algorithm. This program is designed for use in future iterations of the GradLEAP mentorship program.

The mathematical formulations presented offer a solution to the proposed problem; however, additional human intervention may be necessary to identify potential mismatches or suboptimal pairings. Currently, the matching algorithms do not consider responses to free-response questions. We anticipate that the Graduate College can use the model's output and further explore compatibility between pairs by analyzing the free-response answers.

Future analyses could incorporate deep learning and natural language processing techniques to better address this aspect. Alternatively, a keyword search for terms such as "important," "need," or "want" could be employed to flag mentees and ascertain their specific priorities.

Moreover, further analysis is needed to determine the factors that contribute to a successful match. For example, a survey question asking mentees how critical it is for them to be matched immediately could help establish an appropriate threshold. Another area for future investigation is the definition of v_{ijk}. Currently, it is calculated individually for each mentee, but there are inter-mentee factors that should be considered. For instance, if one mentee assigns a score of 5 to every criterion, the v_{ijk} value would be the same as if another mentee had rated each criterion as a 1. However, it is reasonable to assume that achieving an exact match for the first mentee might be more important than for the second.

Our work compares the performance of each model and illustrates the process by which each mentor-mentee pair is created. The Graduate College initially proposed the idea of creating optimized pairings in "waves," given that mentors and mentees continually apply to the GradLEAP mentorship program. This rolling approach to pairing could benefit from the unique characteristics of each solution.

The Hungarian method, which solves a linear programming problem, tends to generate a larger number of pairings with compatibility scores clustered around the mean. In contrast, the Gale-Shapley method produces pairings with greater variation around the mean score.

The solutions presented in Section 2.5 are based on the Cohort 3 data set, where the mean compatibility score is 0.5636. For this data set, 62.80% of the pairs generated by the LP formulation have a compatibility score within 0.2 of the mean, whereas only 54.61% of the pairs from the SMP formulation fall within this range. However, when considering pairs with a compatibility score above 0.8, the SMP formulation yields 17.06% of such matches, compared to 10.92% for the LP formulation.

Based on these findings, we recommend that the Graduate College use the SMP formulation during the early stages of the application process, as it is more likely to produce highly compatible matches. As the number of applicants begins to level off, the LP formulation should then be applied to generate optimal pairings with a mean level of compatibility for the remaining participants. By using both models in waves, the program can optimize compatibility throughout the entire application process.

References

[1] D. Gale and L.S. Shapley, *College admissions and the stability of marriage*, The American Mathematical Monthly 69 (1962), pp. 9–15.

[2] F.S. Hillier and G.J. Lieberman, *Introduction to operations research*, McGraw-Hill, 2015.

Nomenclature

β	age range preference, $\beta \in \{5, 10, 20\}$
$\boldsymbol{J}$	matrix of ones, (m, n)
ℓ	index over uniformly weighted criteria, $\ell \in \{1, \ldots, q\}$
γ	importance rank for age preference, $\gamma \in \{0, 1, 2, 3, 4, 5\}$
$\hat{c}_{ij}$	element of compatibility matrix for assignment problem
$\hat{x}_{ij}$	decision variable for assignment problem
ω_{ik}	Likert scale value that student i gives to criterion k
a_i	age of mentee i
b	branching factor in academic discipline tree
b_j	age of mentor j
C	compatibility matrix, (m, n)
c_{ij}	element of compatibility matrix
d	depth in academic discipline tree
i	index over mentees, $i \in \{1, \ldots, m\}$
j	index over mentors, $j \in \{1, \ldots, n\}$
k	index over weighted criteria, $k \in \{1, \ldots, p\}$
L_j	maximum number of mentees that mentor j is willing to mentor
m	number of mentees
n'	number of mentor slots
N	$\max\{m, n'\}$
n	number of mentors
p	number of weighted criteria
q	number of uniformly weighted criteria
s_i	mentee i

t_j	mentor j
V_k	weight matrix for weighted criterion k, (m, n, p)
v_{ijk}	element of V_k
w	scalar weight for uniformly weighted criteria, 0.01
x_{ij}	decision variable for LP formulation
Y_k	match matrix for weighted criterion k, (m, n, p)
y_{ijk}	element of Y_k
z	objective function
Z_ℓ	match matrix for uniformly weighted criterion ℓ, (m, n, q)
$z_{ij\ell}$	element of Z_ℓ