NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

BC2410

Group Project Report

Finding Love using Linear Programming: Matchmaking & Date Planning

Seminar 2 – Group 3

Name	Matric
Jerome Chew	U1910914B
Lim Guo Quan	U1920769A
Lim Kai Zhe	U2140940E
Chua Gim Aik	U2022142B

Table of Contents

1.0 Introduction to Business Problem

The online dating industry in Singapore is poised for significant growth in the coming years, with revenue projected to reach US$4.23 million by 2023 and the number of online dating app users expected to reach 329.4k by 2027. Based on the statistics from the National Population and Talent Division (Fig 1), there is an increasing number of Singaporeans who meet their partner through online dating apps. However, more than a quarter of online daters reported difficulty in finding someone they were physically attracted to, and over a third found it challenging to find someone with shared hobbies and interests, someone they wanted to meet in person, and someone seeking the same kind of relationship.

After conducting research on popular dating apps like Tinder and Hinge, we have identified the several issues. For instance, due to the 'swiping' mechanic of current dating apps, singles have to spend a lot of time and effort to find a potential match. Since they are exposed to an overwhelming amount of choices, they might also suffer from decision fatigue, affecting their chances of finding a compatible couple. Additionally, these apps lack support for call to actions (such as date planning) which leads to poorer dating experiences. Thus, we will be proposing a solution, to overcome the shortfalls of current apps.

2.0 Proposed Solution

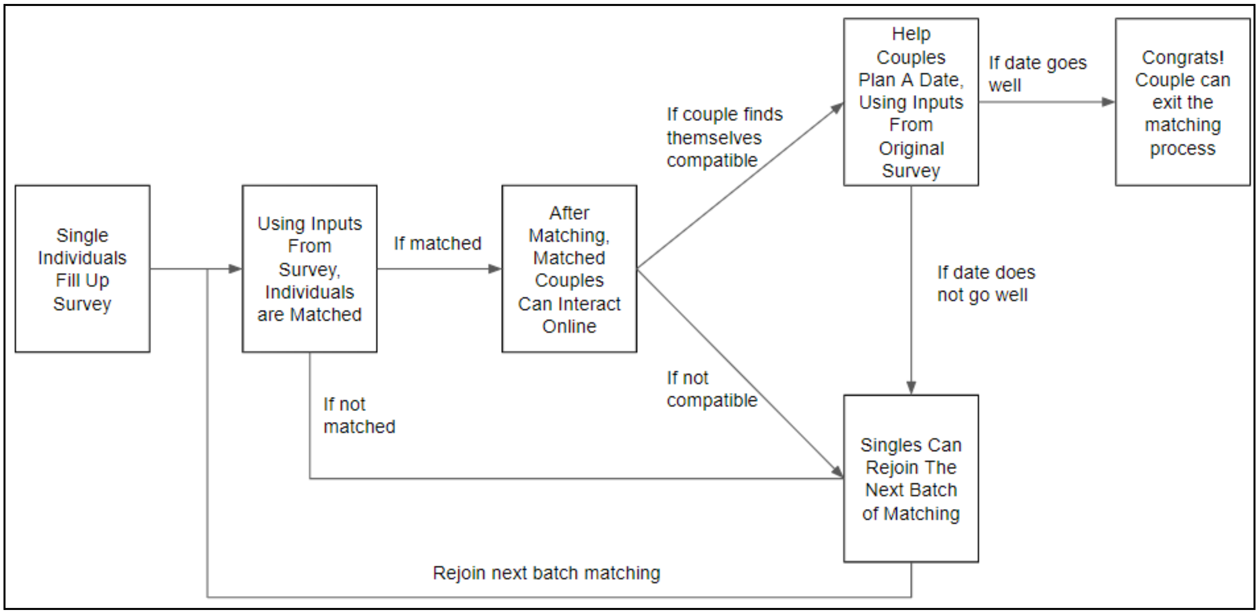
Singapore is currently the most overworked country in the Asia Pacific (SBR, 2022). Given the hectic lifestyle of most Singaporeans, our main target segment would be busy individuals who do not have the time to spend hours on dating apps to find their perfect match. This might include working adults and full-time students who are busy with work/school and may find it difficult to keep up with online dating apps that require continuous involvement and attention. An ideal dating solution should take into account: matchmaking based on compatibility, date planning, and dates should be planned with sufficient time spacing to allow users to focus on managing a small handful of meaningful relationships.

As such, our team has decided to re-model the dating business with an innovative solution that hasn't been tried before. This model is capable of performing the heavy lifting tasks that goes into every good date. It **matches the individuals based on compatibility** and very importantly, **common availability schedules**. It also plans the date for them and even optimises the route to take during the date. This way, participants can focus on the dating experience itself rather than the administrative details. This aspect is grossly overlooked by traditional online dating apps.

2.1 Workflow Of Business

It is important to note that our matchmaking service would take a 'batch dating' approach. Matching will be done weekly. In each batch, we would partition our participants into groups and attempt to match as many people as we can while still ensuring high compatibility for every match within each group. Each week, everyone is only allowed one match and there might be cases where individuals are not matched due to compatibility and availability issues. During date planning, it is important to understand that the date we planned for our couples is merely a suggestion on what we believe the couple might enjoy most. Couples can either follow our itinerary or simply take inspiration from it. Regarding the monetization of our business, after an initial trial period for our service, we could charge singles a monthly subscription rate. We could also get advertising fees from local attractions for featuring them in our itinerary planner.

The diagram below shows how our service works.



3.0 Model Formulation

We will be presenting 3 models that we believe will help transform the dating scene. Namely, we will be exploring how linear programming can solve (1) the matchmaking problem, (2) date planning and finally, (3) route planning for dates

The data we would be using for our models have been collected from 62 fellow NTU students who are also single and actively finding love. The attractions we will be using in our date & route planning models are 50 randomly selected popular attractions for couples to visit in Singapore. Both the student survey and attractions data can be found in the *appendix* section.

3.1 Matchmaking Problem

3.1.1 Introducing: the Matching problem & Objective function

The first problem that we must address is the **matching problem**. Given a group of love-finding participants of size ***N***, how can we best pair everyone up such that everyone's preference is fulfilled as much as possible?

With that, our **objective function** is clear:

$$\max \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij} * Z_{ij}$$

$$x_{i,j} \ binary$$

$$\forall i, j \in \{1, 2, ...n\}$$

- Z_{ij} is compatibility score between two people
- x_{ij} represents the binary selection of a pair

Here, we are trying to maximise the overall group's satisfaction by assigning as many compatible pairs as possible.

3.1.2 Computing Compatibility between two people

Next, we have to discuss how we are going to compute Z_{ij}, our **compatibility score**.There are many ways to compute compatibility. For this project, we will make use of three important variables, ***Interest***, ***Traits***, and ***Seriousness.***

3.1.2.1 Computing compatibility using Interest

Research studies have shown that we are more attracted to people who have similar interests to us. To model this idea, we have to find some way to communicate interest similarity between two people.

Our participants have initially recorded their interest levels (score of 1 - 5) in various activity genres, we can transform this data into an array. For example, the array could look like: [5, 4, 1, …], reflecting a score of 5 for adventure, 4 for Animals & Nature, 1 for Art & Culture, and so on.

To compute the similarity between two people's interest arrays, we can apply the Cosine Similarity between the two array vectors. The intuition is that the angle (*cos* θ) between the two vectors signifies the amount of similarity that two vectors have. One can imagine that two vectors with *cos* θ = 1 are perfectly similar while *cos* θ = 0 means that they are perpendicular and are 'perfectly' different. Note here that the *cos* θ will never be negative as we do not allow interest to hold negative value in our project. Cosine similarity can be formulated mathematical as follows:

$$CS_{ij} = cos\ \theta = \frac{I_i \cdot I_j}{||I_i|| * ||I_j||}$$

where I_i represents the interest vector of person i and so forth.

Now we can measure two people's similarity - but how can we be so sure that people want 'perfectly' similar partners? We feel that to some degree, people want some level of dissimilarity as well, as such, we allowed users to include their similarity preference for their partners. This is denoted as $SimP_i$, which is simply a weight from 0% to 100%. To account for similarity/dissimilarity preference, we can tweak our formulation a little:

$$InterestCompatibility_{ij} = SimP_i * CS_{ij} + (1 - SimP_i) * (1 - CS_{ij})$$

Which can be reduced to:

$$2 * SimP_i * CS_{ij} - SimP_i - CS_{ij} + 1$$

$$\forall i, j \in \{1, 2, ...n\}$$

By taking into account a person's level preference for similarity and conversely, their level of preference for dissimilarity, we arrive at a weighted function that determines interest compatibility between two people. Note that this score is a score ranging between 0 and 1 and it is NOT bi-directional as different people have different preference levels for similarity and hence person j's interest compatibility to person i could be different from person i's interest compatibility to person j.

3.1.2.2 Computing compatibility using Traits

People want their ideal partners to have certain traits. From our survey, we have captured ***traits*** and ***traits preference*** data in the form of binary arrays. Individuals have to select their top 10 traits that best represent themselves from a list of available traits. This leaves us with arrays that look like [1,0,1,...] where '1' represents trait present (e.g. humble) and '0' represents trait absent/not representative.

To account for traits compatibility for let's say, person i and person j - we must first check how partner i's traits matches person j's preferred traits and vice-versa. To do this, we can easily apply cosine similarity again to compute how aligned the traits and traits preferred vectors are. The formulation is as follows:

$$TraitsCompatibility_{ij} = \frac{TP_i \cdot T_j}{||TP_i|| * ||T_j||}$$

$$\forall i, j \in \{1, 2, ...n\}$$

where T_j represents the traits vector of person j, TP_i represents the traits preferred (vector) by person i and n represents the total number of participants

We once again note that the above function yields a score from 0 to 1 and is NOT bi-directional since different people have different trait preferences.

3.1.2.3 Computing compatibility using Seriousness

A big part of matchmaking is to matchmake two people with similar reasons for seeking relationship. Imagine the horrors of matching someone who is 0% serious about finding a partner (looking to make friends) with someone who is 100% serious (looking for long-term relationship), just imagine how disappointed both parties would be! We have previously asked participants about their seriousness levels on a scale from 0% to 100%. Using those scores, we can formulate seriousness compatibility as a measure of how similar two seriousness scores are:

$$SeriousnessCompatibility_{ij} = 1 - |S_i - S_j|$$

$$\forall i, j \in \{1, 2, ...n\}$$

where S_i represents the seriousness score of person i and n represents the total number of participants.

Again, this function yields a value from 0 to 1, but this time it IS bi-directional, meaning person i and j will share the same seriousness compatibility score. Observe that the higher the similarity of the seriousness scores of both parties, the higher the seriousness compatibility score.

3.1.2.4 Combining the various compatibility functions

So far, we have covered three ways to compute compatibility between two people. To combine the three compatibility measures, we will allow the participants to assign weights to each of the three factors. This way, individuals can prioritise the factors they really value. For instance, participants may care a lot about seriousness and decide to assign more weight to seriousness compatibility.

This overall weighted compatibility function can be described mathematically as:

$$C_{ij} = w_0 * InterestCompatibility_{ij} + w_1 * TraitsCompatibility_{ij} + w_2 * SeriousnessCompatibility_{ij}$$
$$\forall i, j \in \{1, 2, ...n\}$$

where w_0, w_1, w_2 represents user selected weights for the compatibility functions for *Interests, Traits and Seriousness* respectively.

Note that the sum of all the weights is 1. This weighted function yields a score ranging between 0 to 1 and is NOT bi-directional. In order to truly match two people, we will need a single compatibility score to represent the compatibility of two individuals. This means we have to somehow make the function bi-directional. To do this, we took the conservative assumption that a pair's compatibility is only as good as the lower compatibility score of the two. This makes sense for us since a good match is one where both parties are satisfied. Mathematically, the *true* compatibility score between two people is simply:

$$Z_{ij} = \min(C_{ij}, C_{ji})$$
$$\forall i, j \in \{1, 2, ...n\}$$

where Z_{ij} represents true compatibility between person i and j.

Note that this is the score we will be using in our maximising objective. This Z score is bi-directional, yields a value ranging from 0 to 1 and is the representation of compatibility between two people.

3.1.3 Setting the constraints

In this next section, we discuss the constraints that our matchmaking model must follow.

3.1.3.1 Basic constraints

(1) Each person must only be allowed 1 match at most
(2) Each pair must be mirrored
(3) Noone should pair with themselves

The above constraints are represented in order from (1) left to (3) right as follows:

$$\sum_{j=1}^{n} x_{ij} \leq 1 \quad \forall i, j \in \{1, 2, ...n\}$$

$$x_{ij} = x_{ji} \quad \forall i, j \in \{1, 2, ...n\}$$

$$x_{ii} = 0 \quad \forall i \in \{1, 2, ...n\}$$

3.1.3.2 Age & Gender requirements

We must also respect everyone's age and gender preference for their partners.

(4) Age gap cannot be greater than both parties' acceptable range
(5) Partner's gender must match preference for both individual

The above constraints are represented in order (4) left to (5) right as follows:

$$x_{ij} * |age_i - age_j| \leq gap_i$$
$$x_{ij} * |age_i - age_j| \leq gap_j$$
$$\forall i, j \in \{1, 2, ...n\}$$

where age_i is the age of person i and gap_i is person i's maximum allowed age gap for their ideal partner

$$x_{ij} \leq gender_i \cdot GenderPreference_j$$
$$x_{ij} \leq gender_j \cdot GenderPreference_i$$
$$\forall i, j \in \{1, 2, ...n\}$$

where $gender_i$ is the gender of person i and $GenderPreference_i$ is person i's preferred gender for their ideal partner

For the *age* constraint, X_{ij} serves as the 'activator', telling the model: "if you choose this pair, their age differences must be smaller than the maximum allowed age gap of person i and person j.

For the *gender* constraint, we note that *gender* is a one-hot encoding binary array, meaning that only 1 element in the array will be '1'. Since the gender preference array is also a binary array stating a person's preference for their partners' gender (users can prefer more than 1 gender: *male, female, others*), we can creatively apply the dot product on both array vectors. If the outcome of the dot product is 1, then the gender requirement is satisfied, otherwise, the outcome is 0 and this matching is not permitted.

3.1.3.3 Compatibility Threshold

(6) To ensure that all our matches have some level of quality, all matches must exhibit a compatibility score at least greater than 0.5

$$x_{ij} * Z_{ij} \geq 0.5$$
$$\forall i, j \in \{1, 2, ...n\}$$

This threshold level can be defined as any value the matchmakers see fit. To remove threshold restriction, simply set the threshold value to 0.

3.1.3.4 Common Availability requirement

Our matchmaking model caters to people who are extremely busy. To ensure that our customers' time is respected, every week, our model only matches people **who are able to meet for a date** due to common availability in their schedule. Additionally, some people may only want to take part in dates that are at least X hours long so as to make full use of their time. To satisfy this, participants will also input their desired minimum date duration. This means the model can only pair people who are available to meet for a date **for as long as both parties' desired minimum dating duration.**

To satisfy this constraint, we must find a way to compute the ***Longest Common Contiguous Availability*** between two people's schedules. LCCA, as the name suggests, is the longest time range that two people are available for without any gaps or breaks. Each person inputs their calendar data which is converted to a 2D binary matrix with the rows being the 7 days of the week and the columns being 1-hour time slots between the period of 8 AM to 10 PM. In this case, an element being ‘1’ means that the person is available for that time slot on that day.

For example, let's take Monday's schedule for:

- person i to be [1,1,1,1,1,1,1,0,0,**1,1,1,1,1**]
- person j to be [1,0,1,0,0,1,1,1,1,**1,1,1,1,1**]

Then, the LCCA for Monday for both of them would be the last 5 hour timeslots (from 5 pm to 10 pm) resulting in Monday’s LCCA to be 5.

To find the week’s overall LCCA, we simply find the maximum LCCA across the 7 days as follows:

$$LCCA_{ij} = \max(LCCA_{ij1}, LCCA_{ij2}, ..., LCCA_{ij7})$$
$$\forall i, j \in \{1, 2, ...n\}$$

where d in $LCCA_{ijd}$ represents the day of the week, i and j represents the id of two different person.

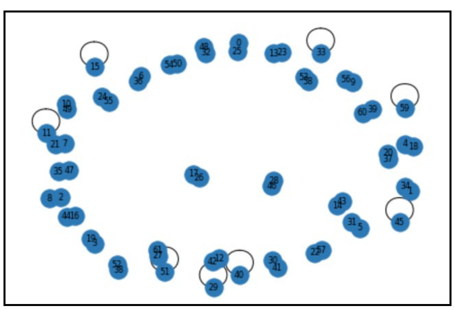
This wraps up our final constraint:

(7) Selected pairs must have common time slots that allows them to meet for as long as both their minimum required dating duration

$$LCCA_{ij} \geq x_{ij} * MinDuration_i$$
$$LCCA_{ij} \geq x_{ij} * MinDuration_j$$
$$\forall i, j \in \{1, 2, ...n\}$$

3.1.4 Results & Model Analysis

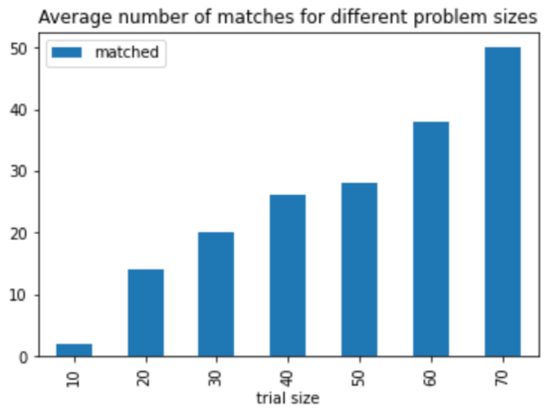
Let’s wrap up the matchmaking model by examining the matching output of our student survey data:

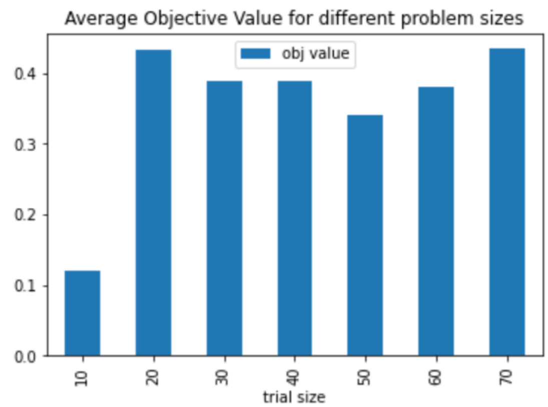


```
Optimal satisfaction: 33.752817147853214
i|j|score|LCCA day|from|to
0 25 0.622222051018559 5 TUESDAY 5 PM 9 PM
1 34 0.6941092690364237 SATURDAY 4 PM 9 PM
2 8 0.519505069659662 3 MONDAY 6 PM 9 PM
3 19 0.6299999999999999 TUESDAY 4 PM 7 PM
4 18 0.7695835425148043 SATURDAY 5 PM 10 PM
5 31 0.6793677222304497 SATURDAY 5 PM 9 PM
6 36 0.5516673211968428 SATURDAY 11 AM 2 PM
7 21 0.5064836003948544 SUNDAY 11 AM 2 PM
8 2 0.5624659907120451 MONDAY 6 PM 9 PM
9 56 0.5948662318919817 SATURDAY 12 PM 4 PM
10 49 0.5950539181869015 SATURDAY 8 AM 2 PM
```

The above cluster graph (left) shows matched pairs as two closely joined nodes, those who did not receive a pairing are marked with a self-loop. As expected, each of the matches here exceeds the required compatibility score of 0.5. Unfortunately, not everyone got matched due to limiting constraints like *age, gender, compatibility and availability.* We observe that the match rate is quite decent as we satisfied 54 people out of 62 by assigning them a compatible partner. Additionally, we also note that the model also considered the optimal meeting time (LCCA) for every pair based on their calendar availability.

3.1.4.1 Matching Efficiency & Quality





To study **matching efficiency** (left figure) and **matching quality** (right figure), we collected and studied the data of the match results on randomly sampled participants of increasing sample size. We collected data from sample size N=10 to N=70. The results are very interesting, **match efficiency** increased with problem size N. This makes sense because the bigger the pool of participants, the more likely it is for people to find compatible matches. On the other hand, **match quality is maintained** relatively stable throughout the various sizes of N. This highlights that our model is doing its job and strategically balancing between satisfying everyone's preferences and making as many matches as possible.

Through this study, we learn that increasing problem size N will be advantageous as it not only performs more matches, the quality of said matches does not suffer. Of course in reality, the matching problem has a NP-Hard complexity and is bounded by the computer's processing capabilities.

3.2 Finding Ideal Date Locations

Now that the couple has been matched, the subsequent model must help them plan their ideal dating itinerary. This date planning model specially caters to our target group — busy individuals with tight schedules. As such, our users do not have to re-enter any inputs at this stage as the data have already been collected at the start. We will make use of ***budget, max_duration, max_distance*** and interest ***arrays*** for both pairs.

We took the conservative approach and assumed the following for a pair's date constraints:

- We take the average interest of the two interest arrays
- We take the lower budget and lower max distance of the two
- We take the maximum allowed date duration to be the LCCA, or one of the pair's preferred maximum duration, whichever is lowest

While aligning with the couple's busy schedule, we want to further push our business proposition of a perfect date by **maximising** the couple's interest levels on each suggested attraction that we provide to them. Our model is as follows:

Objective Function: To maximise couple interest levels

$$\max \sum_{j=1}^{n} x_j I_j$$

where

$$x_j = \begin{cases} 1 & \text{attraction } j \text{ is selected} \\ 0 & \text{otherwise} \end{cases}$$

$$p_{jk} = \begin{cases} 1 & \text{attractions } j \text{ and } k \text{ are selected} \\ 0 & \text{otherwise} \end{cases}$$

$$I_j = \text{interest level of attraction } j$$

We can observe that there is an $\mathbf{I_j}$ variable, and this is an array of the average interest between the couple in regards to a certain category of attraction. For example, if partner A gives the category of **Wellness** a 4 out of 5 score, while partner B gives it a 2 out of 5 score, the average interest tagged to the **Wellness** category will be $\frac{4+2}{2} = 3$ points. This is to ensure that both partners are equally accounted for. The model constraints are as follows:

Constraint 1: At least 1 attraction is suggested

$$\sum_{j=1}^{n} x_j \geq 1$$

Where n is the number of attractions

Constraint 2: Total cost of all suggested attractions less than budget

$$\sum_{j=1}^{n} x_j c_j \leq budget$$

Constraint 3: Total time of all suggested attractions less than time

$$\sum_{j=1}^{n} x_j t_j \leq time$$

Constraints 4 to 6 ensure that if locations j and k are selected, p_{jk} cannot be more than 1

Constraint 4

$$\sum_{j=1}^{n} \sum_{k=1}^{n} p_{jk} \geq x_j + x_k - 1$$

Constraint 5

$$\sum_{j=1}^{n} \sum_{k=1}^{n} p_{jk} \leq x_j$$

Constraint 6

$$\sum_{j=1}^{n} \sum_{k=1}^{n} p_{jk} \leq x_k$$

Constraint 7: If locations j and k are selected, the distance between them is less than $distance$

$$\sum_{j=1}^{n} \sum_{k=1}^{n} p_{j,k} * D_{j,k} \leq distance$$

Where D is the distance between j and k

Bringing special attention to constraints 3 and 7, we are prioritising the time of the couple, by ensuring the planned activities do not exceed the maximum allowed duration of the date, and that the maximum distance between each suggested location is adhered to. In this sense, we are allowing busy couples to spend sufficient time on their dates while still respecting their preferred date duration and travel distance. The final output will be a set of suggested attractions that the couple can visit. For a sample output visit *Figure 3* in the Appendix section.

3.3 Route Planning

Once we have determined the ideal attractions for our couples to visit during their date, we can now help them plan the route to take for their date. Since our couples are likely to be extremely busy individuals, we would want them to take the shortest route between attractions to save travelling time. We have also allowed the couple the choice of their starting location to cater to the preferences of our couple. Since this is an application of the travelling salesman problem (TSP) we used the Miller-Tucker-Zemlin (MTZ) formulation. We first label the attractions with the number 1 to N, where N is the total number of attractions that our couple would be visiting for the date. However, we have decided to only solve the TSP when our part 2 gives us 3 or more locations. This is because when there are only 1 or 2 locations, their shortest path is extremely intuitive. This would help us to save time and resources when our project scales up. The model is as follows.

Objective Function: To minimize the total distance travelled

$$\min \sum_{i=1}^{N} \sum_{j=1}^{N} d_{ij} x_{ij};$$

where

$$x_{ij} = \begin{cases} 1 & \text{the path goes from attraction } i \text{ to attraction } j \\ 0 & \text{otherwise} \end{cases}$$

$$d_{ij} = \text{the distance between attraction i and attraction j}$$

subject to:

Constraint 1: Only 1 path goes to attraction j

$$\sum_{i=1}^{N} x_{ij} = 1 \qquad j = 1, \ldots, N;$$

Constraint 2: Only 1 path comes from attraction i

$$\sum_{j=1}^{N} x_{ij} = 1 \qquad i = 1, \ldots, N;$$

Constraint 3: If there is a path from i to j, Ui comes after Uj

$$u_i - u_j + N x_{ij} \leq N - 1 \qquad 2 \leq i \neq j \leq N$$

$$1 \leq u_i \leq N - 1 \qquad 2 \leq i \leq N;$$

$$u_i \in \mathbf{Z} \qquad i = 2, \ldots, N;$$

Where u_i is a dummy variable indicating the order which the attraction is visited $\quad i = 2, \ldots, N;$

Constraint 4: Xij is binary

$$x_{ij} \in \{0, 1\} \qquad i, j = 1, \ldots, N$$

Constraint 5: You cannot travel to the same attraction from the same attraction

$$x_{ii} = 0 \qquad i = 1, \ldots, N;$$

Upon solving the model using RSOME, we are able to obtain X_{ij} which shows us the ideal paths to take. Using a simple algorithm, we can then determine the optimal order of attractions to visit from X_{ij}. This path would then be displayed to the user. For sample output visit *Figure 4* in the Appendix section.

4.0 Future Improvements

4.1 Matchmaking Problem

The first most obvious improvement is to **include the use of other variables/factors** that people look out for when dating like *language, religion, job, income, education, etc.* This would allow the compatibility matching to become more robust and hence create higher quality matches. Most of the factors yet to be considered can easily be added using the same methods that we have used for the existing variables in the current model. In general, adding meaningful constraints will improve the quality of matches at the cost of match efficiency. This also means that one must add constraints with caution as adding unnecessary ones would dilute the match quality and also drastically impact match efficiency.

Another improvement we can consider is strategically grouping people together based on some similar attributes. This could mean grouping people by *age, interest, schedule, etc.* In practice, applying the model onto strategically grouped participants will yield better matching results as some constraints are being relaxed by the nature of the grouping. In such cases, one might expect better matching quality and volume.

4.2 Finding Ideal Date Locations

Currently, our attraction list contains a limited number of attractions (50) due to the Open Route Service API's restrictions. Thus, we are unable to consider all the possible attractions in Singapore. As a future improvement, we could possibly expand our attraction lists to consider more attractions and this would allow for better date itineraries.

4.3 Route Planning

Using the MTZ formulation, the optimal route is generated as a loop. So we are making the underlying assumption that the couples would want to end the date back at their starting location, as the case with most guided tours. However, this may not be the case in all situations. Thus, in order to finetune our route planning, we could possibly plan the route based on the ideal start and end location for our couples. In addition, our current formulation is based on the assumption that our couples would be driving. However, in the future, we could possibly include other modes of transport such as the MRT or bus.

5.0 Conclusion

In conclusion, we believe that by building an all-in-one dating solution, not only would we be able to cater to the needs of busy individuals with tight schedules, we would also be able to address some of the pitfalls of current applications. With the use of our matching optimization model in part 1, we have made our solution more time efficient than current swiping apps like Tinder and Hinge, as our users would not have to spend hours on our application, trying to swipe and find their perfect match. Based on our optimization algorithm, we also provide our users with very compatible matches, allowing them to focus their efforts only on compatible partners. With part 2 and part 3 of our model, we also go one step further for our users, helping them plan the perfect date that would be enjoyed by both parties. This also helps to save time for our couples as they do not have to spend time researching on places to go, coordinating their schedules to find a suitable date, as well as planning the route for their trip. Since no current dating application provides this service, we believe that this provides us with a first mover advantage in this field, allowing us to gain a competitive advantage in the market. Therefore, we believe that with this matchmaking & dateplanning service platform, we would be able to capture and profit from the growing online dating segment in Singapore.

End.

Appendix

Data Dictionary

We collected the following data using google forms and with the consent of our fellow NTU peers. We have anonymized participants by replacing their names with numeric ids. We have also pre-processed the .csv file to be in the correct data format for easy importing.

Participants Dataset

Age: Age of participant

Gap: Desired age gap of partner

Gender: Array of binary variables, indicating gender. [Male, Female, Others]

Preference: Preference of partner gender [Male, Female, Others]

Interests: Array containing the interest level of participant of the following categories: Adventure, Animals and Nature, Art and Culture,Fitness, Foodie, Recreational, Sightseeing, Wellness

Seriousness: Seriousness of the participant in finding a relationship on a value range from 0% (casual dating) to 100% (serious dating)

Similarity: How similar the participant would like their partner to be on a value range from 0% being totally different to 100% being totally similar

Rankings: How the participant would weigh the following factors: Interest + Preference for Similarity, Traits, Seriousness

Traits: The top 10 traits that participants feel represents them. The traits are namely: Ambitious, Authentic, Caring, Cheerful, Committed, Confident, Creative, Easy-going, Energetic, Funny, Generous, Honest, Humble, Independent, Intelligent, Kind, Loyal, Mature, Open-minded, Optimistic, Outgoing, Patient, Practical, Respectful, Responsible, Romantic, Social, Supportive, Thoughtful, Trustworthy.

Traits_preference: The traits the participant values in their partner

Budget: The budget of the participant on an initial date

Min Duration: The minimum duration the participant would like to spend on a date

Max Duration: The maximum duration the participant would like to spend on a date

Max Distance: The maximum distance the participant would like to travel between attractions on a date

Calendar: A binary 2d matrix where each row corresponds to each day of the week (Mon-Sun) and each column represents a specific 1-hour time slot. Obtained from participant's calender schedule

Attractions Dataset

Location: Name of attraction
Type: Interest type of attraction
Cost: Cost of activity per pax
Duration: Time spent on activity
Lat: Latitude of attraction
Lon: Longitude of attraction

Figures

Figure 1 (Chart: Straits Times Graphics, Source: National Population And Talent Division)

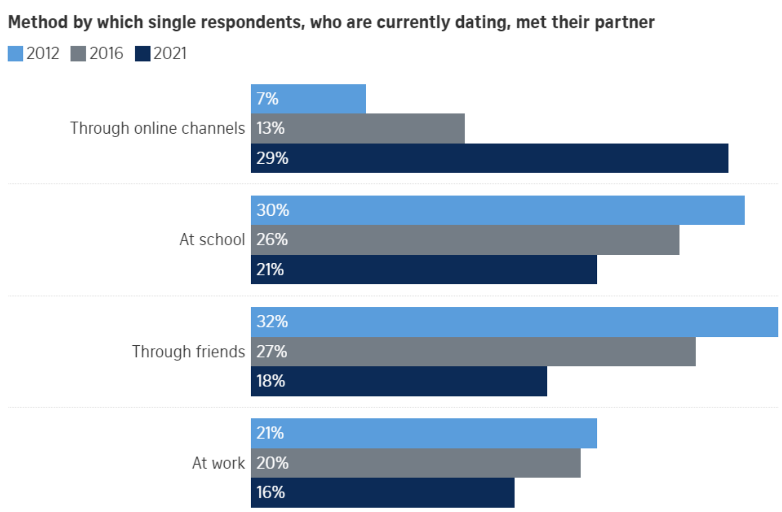


Figure 2 (Source: Pew Research Center)

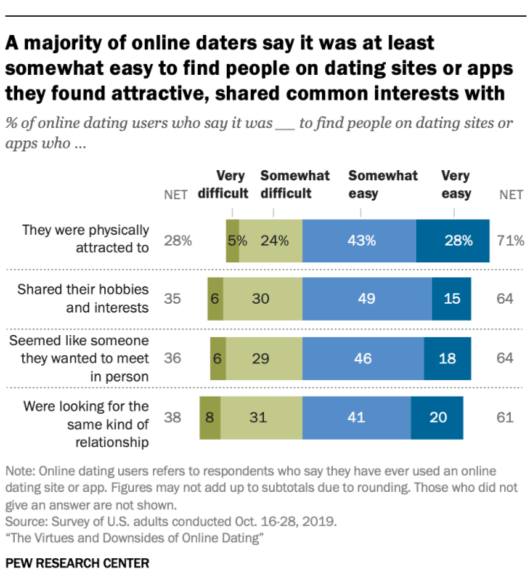


Fig 3: Example of date planning using a pair matched by the matchmaking model

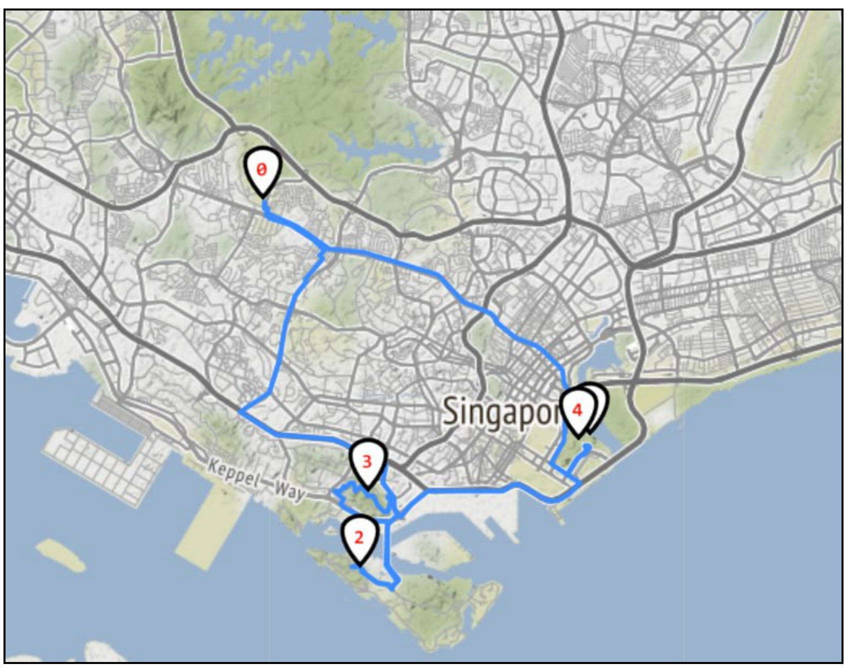
	id	partner	LCCA	interests	min_duration	max_duration	budget	max_distance
12	13	23	4.0	[3, 4, 4, 5, 1, 1, 5, 4]	2	7	50	28
21	23	13	4.0	[3, 2, 3, 5, 2, 2, 2, 5]	2	4	40	31

Theres are the recommended activities for the above pair:

- Date duration should be 4, since LCCA and partner 13's max_duration preference is the lowest
- Budget is 40
- Max distance between two locations is 28 km

	No.	Location	Type	Cost	Duration (Hours)	Lat	Long	Interest Level
0	14	Gallop Stable (Horse Riding)	Animals and Nature	5.0	0.5	1.333475	103.797447	3.0
1	39	Gardens by the Bay (OCBC Skyway)	Sightseeing	5.0	1.0	1.283199	103.866697	3.5
2	40	SkyHelix Sentosa	Sightseeing	8.1	1.0	1.255099	103.818120	3.5
3	42	Singapore Cable Car	Sightseeing	14.0	0.5	1.271359	103.819782	3.5
4	43	Gardens by the Bay (Supertree Grove)	Sightseeing	0.0	1.0	1.282155	103.864324	3.5

Fig 4: Output of route planning for planned date above




```
Ideal path is [(0, 4), (1, 3), (2, 0), (3, 2), (4, 1)]
```

References

Singapore named most overworked country in APAC. (2022, June). Singapore Business Review. https://sbr.com.sg/hr-education/in-focus/singapore-named-most-overworked-country-in-apac

Vogels, E. A. (2020, February 6). 10 facts about Americans and online dating in 2019. Pew Research Centre. https://www.pewresearch.org/fact-tank/2020/02/06/10-facts-about-americans-and-online-dating/

Goh, Y.H. (2022, October 10). More couples meeting through online dating sites or apps: Survey. The Straits Times https://www.straitstimes.com/singapore/more-couples-meet-through-online-dating-sites-or-apps-over-the-years-survey