Finding LOVE with Linear Programming

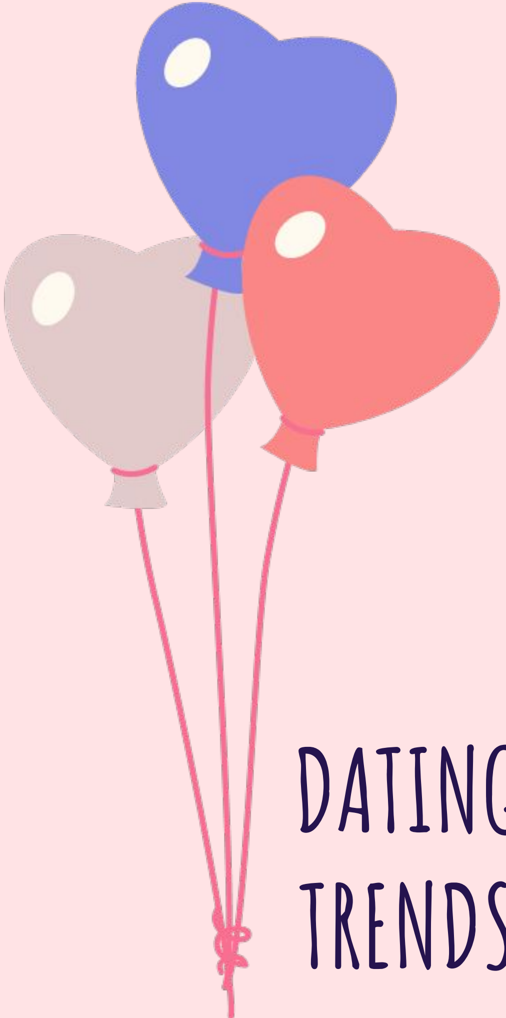
Matchmaking & Date Planning

BC2410 PROJECT

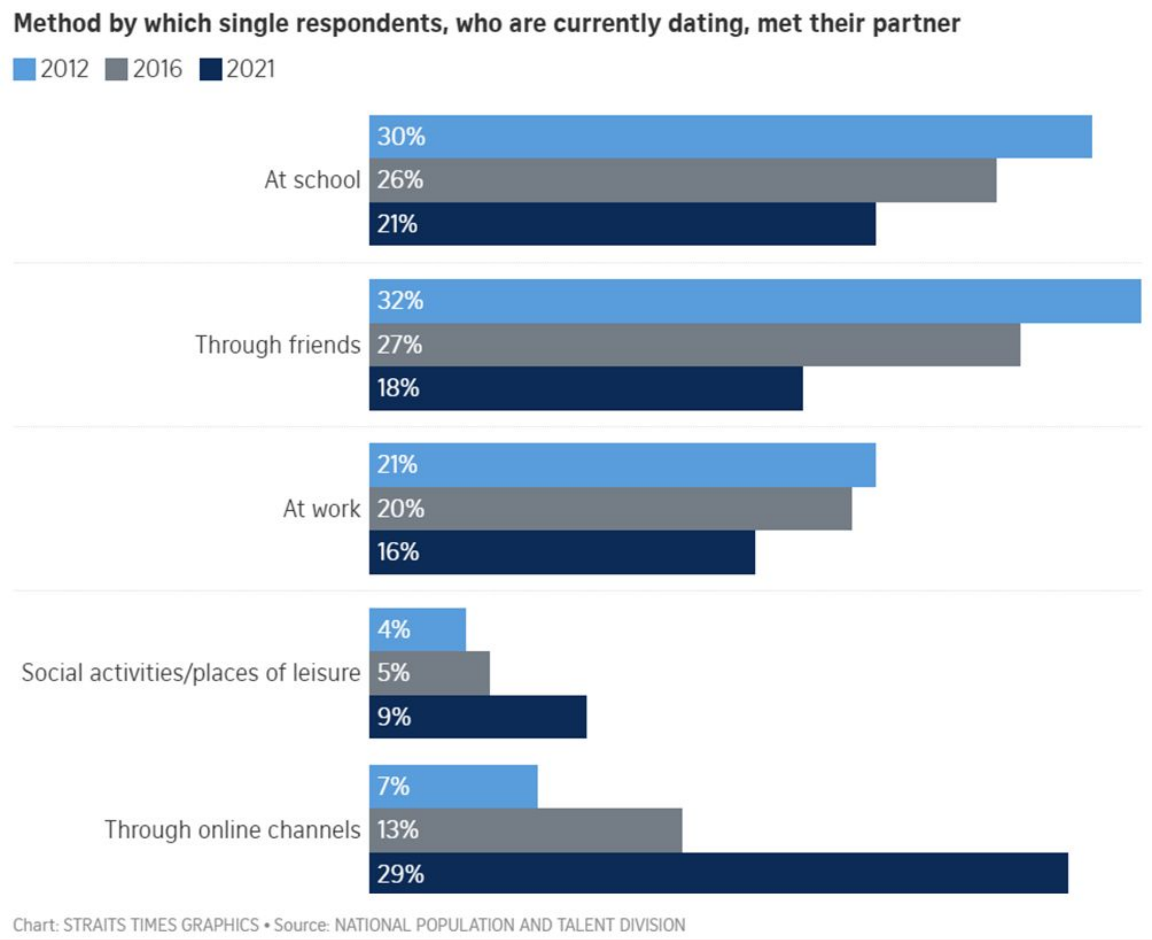
Chua Gim Aik
Jerome Chew
Lim Guo Quan
Lim Kai Zhe

Have you seen these logos before?

Yes…… all of them are online dating apps!



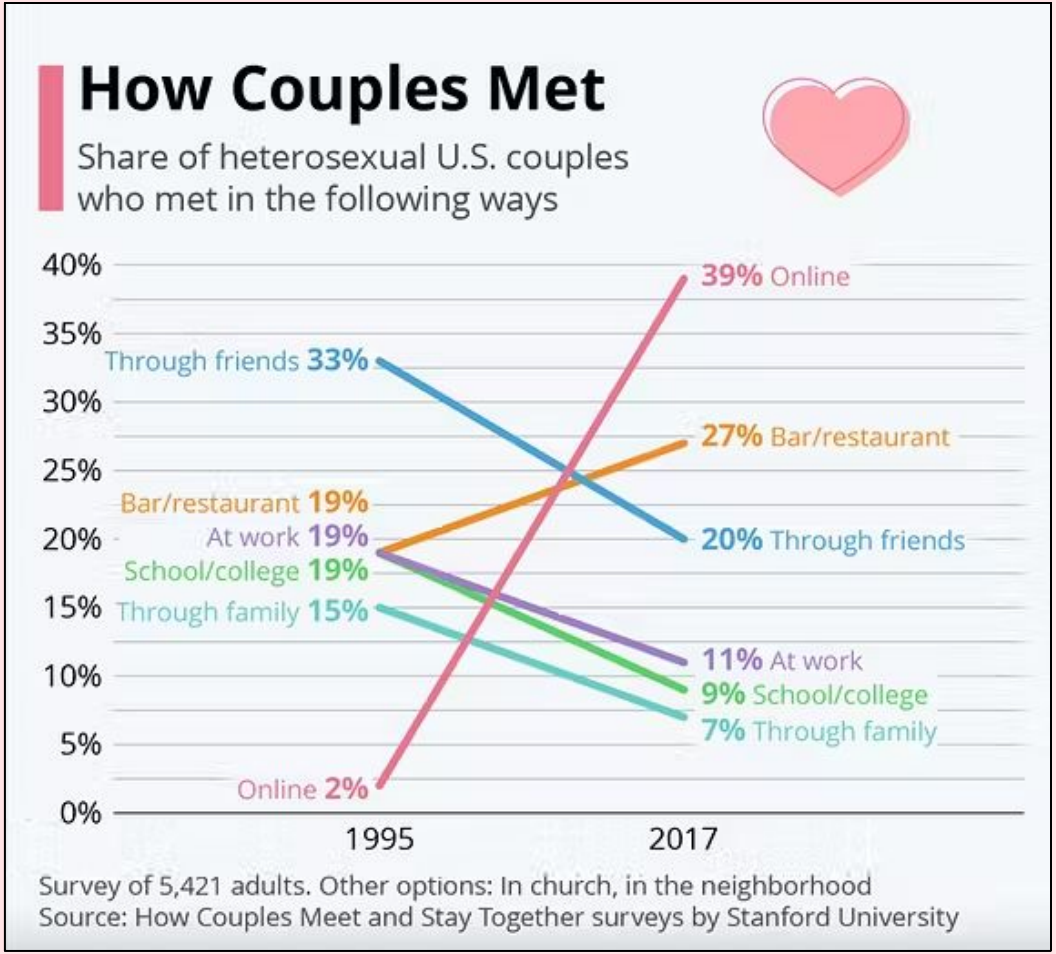
DATING TRENDS



Revenue in the Online Dating industry in Singapore is projected to reach US$4.23 millions in 2023.

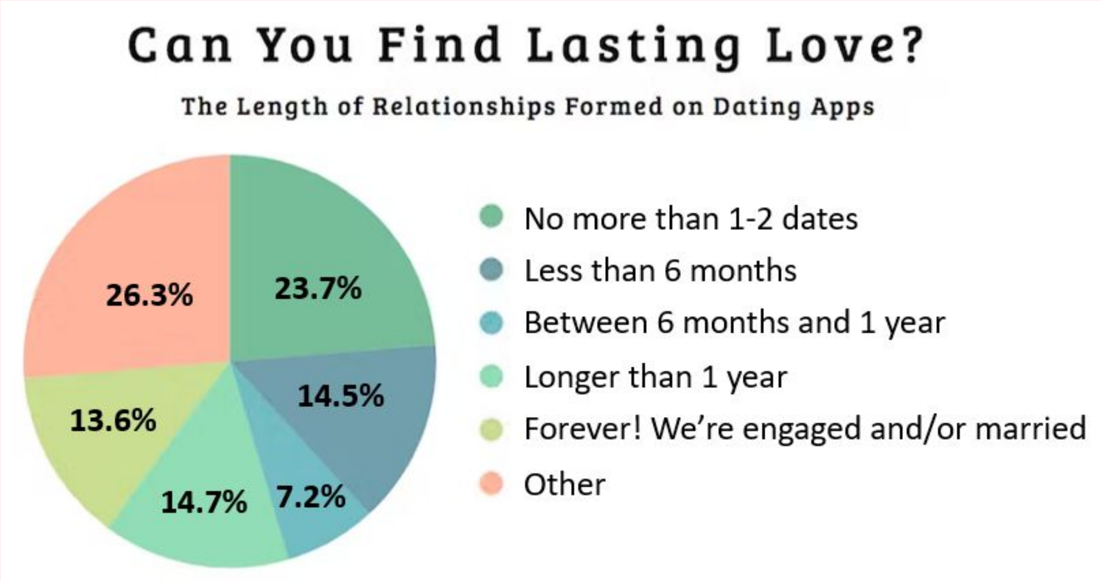
The number of online dating app users is expected to amount to 329.4k users by 2027.

(Source: Statista, Digital Market Insights, eServices, Dating Services)

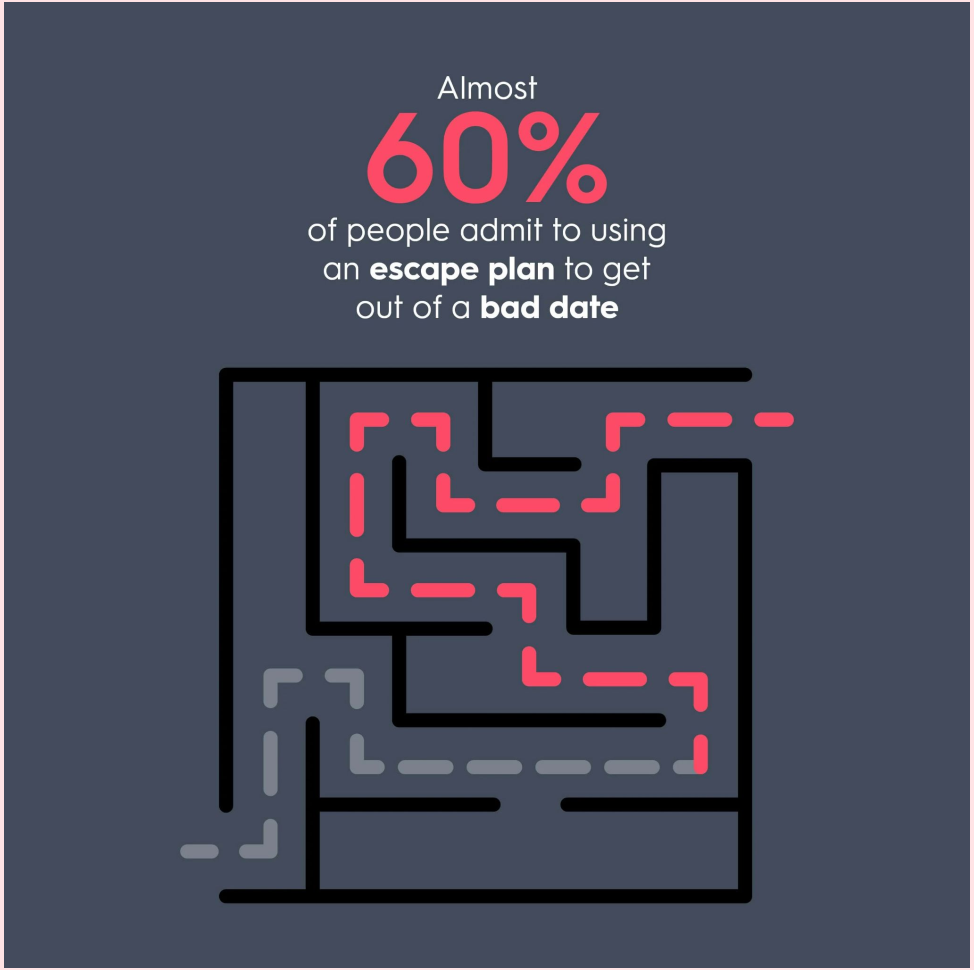

How Couples Met
Share of heterosexual U.S. couples who met in the following ways
40%
35%
30%
25%
20%
15%
10%
5%
0%
Through friends 33%
Bar/restaurant 19%
At work 19%
School/college 19%
Through family 15%
Online 2%
39% Online
27% Bar/restaurant
20% Through friends
11% At work
9% School/college
7% Through family
1995
2017
Survey of 5,421 adults. Other options: In church, in the neighborhood
Source: How Couples Meet and Stay Together surveys by Stanford University


39% of couples meet and stay together through online dating!

However, only 13.6% of the relationships formed on dating apps ended up marrying.



I HAD A BAD DATE :(



Opportunity Areas

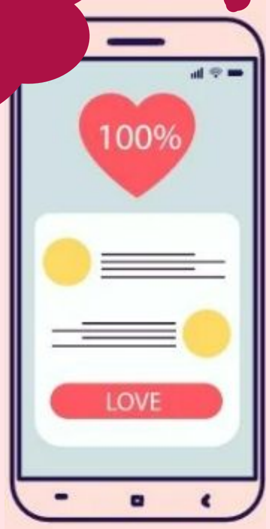
Current Dating Application Scene

Popular dating applications like Tinder and Coffee Meets Bagel provide preference matching via an intuitive swipe system.

Why does a dating fail?

Does not match with my impression

Not suitable



Not an ideal dating location for both parties

I don’t know where to go in Singapore

We are here to solve your problem!

Who are we? What we do?

TRUE LOVE

A new company providing comprehensive preference matching

Help people find meaningful connections through our dating services

We provide a perfect package of services.

Personalized Matchmaking

• find the best matches for all of our clients
• by using optimization algorithms

Plan a Date

• choosing the right activities and restaurants

Create a Complete Itinerary

• navigation in Singapore can get messy
• provide a route planner based on the shortest path

PERFECT DATE!

Matchmaking with Mathematics

Who to date?

The matchmaking problem

Instead of mindless ‘swiping’, and allowing a user to have large number of matches, we want participants to focus on building a handful of meaningful relationships.

To achieve this, we need to solve **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?

Objective function

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

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

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

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

- The goal here is to maximize the overall compatibility for **every assigned pair**

How do we compute compatibility score between 2 people?

Factors we considered

- **Age**
- **Gender**
- **Interests**
- **Preference for similarity** (Do you believe in ‘opposites attract’?)
- **Traits**
- **Seriousness** (Intention for seeking relationship)
- **Values ranking** (Which factors are more important to you?)

Other factors to consider

- Language
- Income
- Education
- Job
- Height/other physical attributes
- Preference for smoking/alcohol
- Personality
- Moral Values
- Religion
- etc.

Computing compatibility

We will look at three measures for measuring compatibility between two people:

Interests

Traits

Seriousness

Finding Compatibility by Interest

Users will indicate their interest levels (1-5) for each of the following interest genre:

Adventure | Wellness | Sightseeing | Animals & Nature | Art & Culture | Food | Fitness | Recreational

This creates 'Interest' vector arrays for each user:

E.g. [5, 4, 3, 4, 1, 1, 5, 2]

Can we find the similarity between two sets of interest arrays?

Finding Compatibility by Interest

Yes! We can use cosine similarity to determine 'similarity' between two vectors

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

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

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

We can account for similarity/dissimilarity preference by …

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

where $SimP_i$ refers to the similarity preference for person i

Finding Compatibility by Traits

Users will indicate their top 10 traits and preferred traits for their partners in a binary vector array:

Funny | Loyal | Independent | Romantic | Thoughtful **, … , etc.**

This creates 'Traits' and 'Traits preferred' vector arrays for each user:

E.g. **I am …** [0, 1, 1, 0, 1, …]

I want [1, 1, 0, 0, 0, …]

Finding Compatibility by Traits

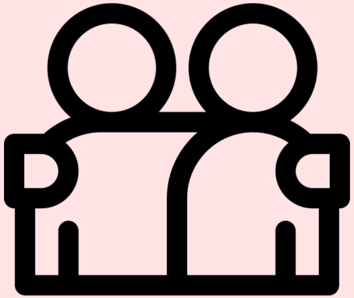
Once again, we use cosine similarity to determine traits matching

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

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

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

Finding Compatibility by Seriousness

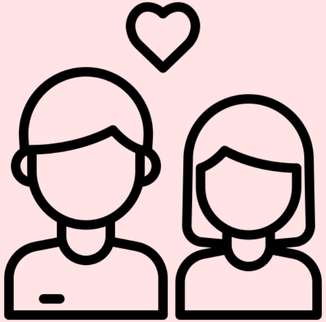


0%

I am looking to make a friend only, no relationship

50%

I am not particularly looking for anything



100%

I am looking to find a partner through this

Finding Compatibility by Seriousness

We will match people based on how similar their '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.

Consolidating all the compatibility functions

You decide which compatibility factors are important!

The formulation is as follows:

$$C_{ij} = \underline{w_0} * InterestCompatibility_{ij} + \underline{w_1} * TraitsCompatibility_{ij} + \underline{w_2} * SeriousnessCompatibility_{ij}$$

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

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

- The resulting value is a score ranging between 0 and 1.
- C_{ij} is NOT bi-directional. That is, $C_{ij} \neq C_{ji}$

Consolidating all the compatibility functions

True compatibility score

$$Z_{ij} = \underline{\min}(C_{ij}, C_{ji})$$

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

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

- Z_{ij} is bi-directional. That is, $Z_{ij} = Z_{ji}$
- The resulting value is a score ranging between 0 and 1.
- **This value is the compatibility score that we are trying to maximise in our model for ALL our participants**

The Matchmaking Model

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

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

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

(1) ONE match per person (at most)

$$\sum_{j=1}^{n} x_{ij} \leq 1$$

(2) Pairs must be mirrored

$$x_{ij} = x_{ji}$$

(3) You cannot pair with yourself

$$x_{ii} = 0$$

The Matchmaking Model

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

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

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

(4) Age gap cannot be greater than both parties' acceptable range

$$x_{ij} * |age_i - age_j| \leq gap_i$$

$$x_{ij} * |age_i - age_j| \leq gap_j$$

(5) Gender must match preference for both individual

$$x_{ij} \leq gender_i \cdot GenderPreference_j$$

$$x_{ij} \leq gender_j \cdot GenderPreference_i$$

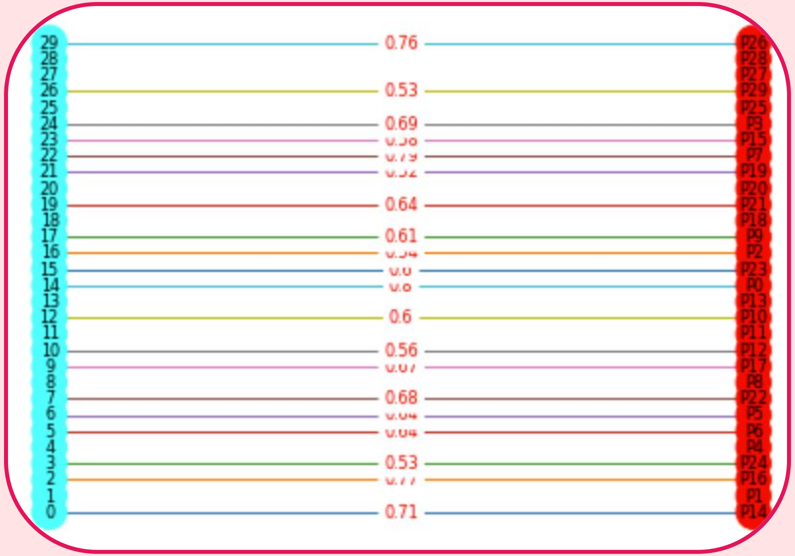
Genders: M | F | Others
(Binary array) e.g. "Male" = [1, 0, 0]

(6) Compatibility score must be at least 0.5 (for quality matching)

$$x_{ij} * Z_{ij} \geq 0.5$$

The Outcome

Example of an optimal matching solution for an (N=30) problem



Not everyone gets matched due to *age, gender* and *compatibility* restrictions.

Finding Ideal Date Locations

After Finding the Perfect Partner… Let’s Plan your Date!

Maximise Interest for Each Suggested Date Location

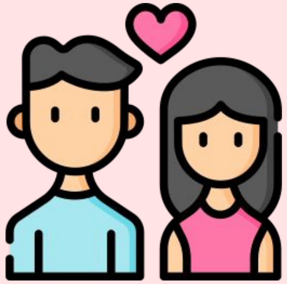
We want both partners to enjoy themselves to the max!

Allow the Couple to Determine Constraints

To get a better sensing of the couple’s preferences!

A Perfect Date Set-Up

We provide 4 locations for the perfect date for the lovely couple!



Finding Ideal Date Locations

Couple Determined Variables

Interest Level	Budget	Day	Time	Distance
Couple indicates interest levels for each category of activity	Couple indicates desired maximum budget	Couple indicates which day of the week they are both free	Couple indicates maximum duration of date in hours	Couple indicates maximum distance between every suggested location

Finding Ideal Date Locations

Couple Determined Variables

Interest Level

Couple indicates interest levels for each category of activity

Finding Ideal Date Locations

Couple Determined Variables

Interest Level

KF1 Karting	Adventure
MacRitchie Tree Top Walk	Adventure
Katapult Trampoline Park	Adventure
Skyline Luge	Adventure
Wild Wild Wet	Adventure
Elements Wellness	Wellness
G.Spa	Wellness
Joya Onsen Cafe	Wellness
Syoujin Massage	Wellness
Spa1 Wellness	Wellness
Sunbathing at Siloso Beach	Wellness

Average Interest Score: 3.5

Average Interest Score: 5

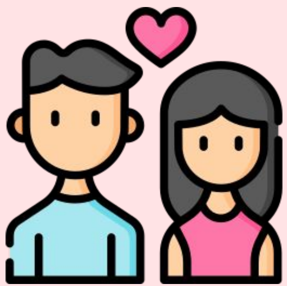
Finding Ideal Date Locations

Couple Determined Variables

Interest Level | Budget | Day | Time | Distance

Let’s spend max **$100**! And I know we are both free on **Thursday**!

Yup! Let’s have a loooong date of **10 hours**! I will be driving, so maximum distance can be **20km**!!



Finding Ideal Date Locations

Main Goal: Maximise couple's interest levels

Decision Variables

- *x stands for choosing a location, which is a binary variable*
- *p stands for choosing two locations, which is a binary variable (explained later on)*

LOP Equation:

$$\max \sum_{j=1}^{n} x_{i,j} I_j$$

Where ***i*** is the selected day, ***j*** is the activity number, ***n*** is the number of locations we have in our dataset and **I** is interest levels for each activity

Finding Ideal Date Locations

Constraints:

- Activity is only valid for selection if available on selected day: $x_j \leq a_{i,j} \quad i = \{Day\}, \forall j \in \{1, 2, \ldots, n\}$
- Only 4 activities can be selected in one day: $\sum_{j=1}^{n} x_j = 4$
- Total cost is less than user determined budget: $\sum_{j=1}^{n} x_j c_j \leq budget$
- Total time is less than user determined time constraint: $\sum_{j=1}^{n} x_j t_j \leq time$

Finding Ideal Date Locations

For every pair of locations, *j* and *k*:

$$p_{j,k} \geq x_j + x_k - 1$$

$$p_{j,k} \leq x_j$$

$$p_{j,k} \leq x_k$$

$$\boxed{p_{j,k} * D_{j,k} \leq distance}$$

$$\forall j, k \in \{1, 2, \ldots, n\}$$

The *p* decision variable is used here. If $p_{j,k}$ is 1, it means both location j and k are selected. These first 3 constraints make sure that if locations j and k are to be selected, $p_{j,k}$ will be 1.

If $p_{j,k}$ is 1, it means both locations j and k are selected. This constraint makes sure the distance between locations j and k are less than the user determined ***distance***.

Finding Ideal Date Locations

Final Output

```
Your ideal date would include these locations:
Gardens by the Bay (OCBC Skyway)
SkyHelix Sentosa
Singapore Flyer (Singapore Sling Experience)
Botanic Gardens
The total cost of your date would be: $95.20
```

Finding the shortest path between attractions

Next Steps

After determining the optimum travel itinerary, we would then ask if the user would like to choose the starting location of the date

We would then find the most efficient way to travel between those attractions using the travelling salesman MTZ formulation.

If user chooses starting location

- Suggest optimal route based on that starting location

If user has no preference

- Suggest a generalised loop

Shortest Path Formulation

Objective Function

Objective Function: To minimize the total distance travelled

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

where

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

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

*Following Miller-Tucker-Zemlin Formulation

Shortest Path Formulation

Constraints

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

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

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

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

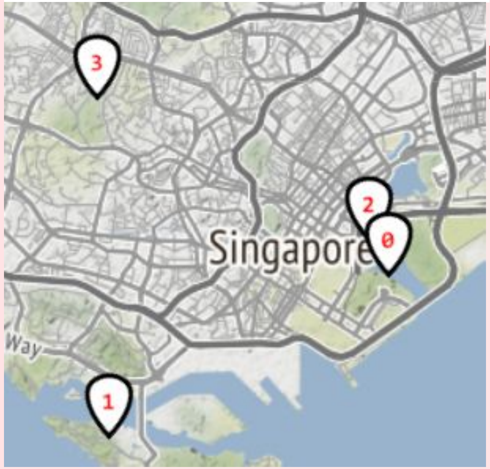
$$x_{ii} = 0 \qquad i = 1, \ldots, N;$$

*Following Miller-Tucker-Zemlin Formulation

Finding the shortest path between attractions

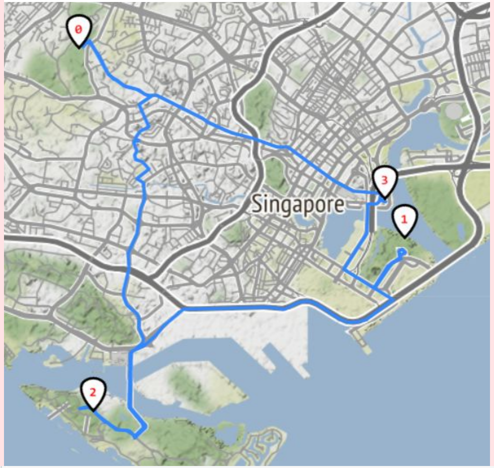
Chosen Locations

	No.	Location	M
0	18	Gardens by the Bay (OCBC Skyway)	
1	19	SkyHelix Sentosa	
2	20	Singapore Flyer (Singapore Sling Experience)	
3	26	Botanic Gardens	



If user chooses starting location to be botanic gardens

	No.	Location
0	26	Botanic Gardens
1	18	Gardens by the Bay (OCBC Skyway)
2	19	SkyHelix Sentosa
3	20	Singapore Flyer (Singapore Sling Experience)




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

Desired Impacts of Our Solution

Business Level

- By creating an all in one application, we hope to be able to attract many users to use our platform.
- Capture and profit from the growing online dating segment in Singapore.

National Level

<u>Promote Local Tourism</u>

- Through our application, we would be able to draw visitors to less well known attractions in Singapore, promoting local tourism and revenue

Future Possible Improvements

Matching Problem

Model requires individuals to know themselves well. Model will suffer in performance if they don't.

- To rectify this, we can consider a model that factors in a user's confidence in describing themselves
- We can also consider more attributes for the model

Attraction Selection

Activities might be subject to present-day weather conditions

- Implement a contingency wet-weather plan
- Ensure the new suggested locations follow the same constraints

Create different itineraries for different days of the week

Shortest Path Finder

- Factor in the homes of both parties to find the optimal starting location
- Create routes based on local transport options (E.g MRT, bus) using Singapore's open route API 'OneMap'

Thank You

Any Questions?