BC2410/2411 - Prescriptive Analytics: From Data to Decision AY22/23 Semester 2

Project Report

Project Title: **Itinerary Planner**

Date: **16 April 2023**

Instructor: **Asst. Prof. Tang Qinshen**

Seminar Group: 3	**Team: 4**
Chan Shao Jing	*U1921314E*
Jain Karisha	*U2111392H*
Low Jin Teng Jackson	*U1922836D*
Yeo Galvin	*U2010712B*

Table of Contents

1. Business Problem

As human beings, it is in our innate nature to have a strong curiosity to explore new places, experience new cultures and immerse ourselves in novel environments. As the world becomes increasingly globalized, it is now easier than ever to travel to any part of the world.

Nonetheless, to ensure an enjoyable trip filled with fun activities, every trip requires extensive planning which may take up more time than the trip itself. According to a study published in the Applied Research in Quality of Life journal, the most significant increase in happiness is associated with the uncomplicated act of preparing and organizing a vacation. The study also found that only vacationers who reported feeling "very relaxed" during their trip experienced a boost in happiness after the vacation. For these individuals, the positive effect on their happiness lasted for more than two weeks following the trip, before returning to their normal level of happiness (Nawjin, 2010). This research further affirms the importance of detailed planning for a rewarding vacation.

Further studies have also shown that 69 percent of Americans plan their trip during working hours (Russo, 2016). This reinforces how time- and effort- consuming planning a trip can be and the strong need for an efficient and user-friendly trip planner. Therefore, this project has been designed to attempt to solve the following question:

"How can we minimise the time and effort to plan a trip efficiently and effectively while maximising the user-specified objectives and requirements?"

In the current market with rapidly advancing machine learning and artificial intelligence (AI) technologies, there are many off-the-shelf products that aid travellers in planning their itinerary. However, these solutions mainly aim to solve only one area while lacking in others. For example, Notion AI plans a trip for you based on the city you are going to, recommending certain itineraries based on certain interests that the user has. Conversely, the AI is lacking in situations where the user has a specific list of activities they would like to do, with a certain set of activities they must visit and would like to fit as many activities as possible within the number of days that they have for the trip to obtain a minimal distance travelled. The current Notion AI product will only suggest "cool destinations" that users may be interested in but does not allow much flexibility for structured user inputs. Notion AI also seemed to produce unrealistic travel plans ignoring the distances between destinations. Recommendation engines also do not seem to be high in demand as a simple search of "top 30 things to do" in a certain city would give similar outcomes. Thus, this project aims to solve the problem statement while considering the shortfalls of itinerary planners in the market.

2. Strategies & Feasibility

The intended outcome of this project is to achieve an optimal trip itinerary with several user inputs. The model would be targeting users who would like to receive a planned itinerary with minimal user research on what they would like to do with an option to have preferences on which of the activities are a “must-do” for them.

2.1 Front-End Strategy

The model will require the following user inputs:

1) An excel sheet of the list of shortlisted activities with details such as address of the activity, cost, estimated duration of activity, and whether it is a “must-do”
2) An excel sheet of the list of shortlisted accommodations with details such as address of the accommodation and cost per night
3) The start and end dates of the trip
4) The daily starting and ending time desired

2.2 Back-End (Model) Strategy

With the inputs from the users on the front-end, we would want to choose a model strategy that can fulfil the following factors:

1) Optimize the number of activities chosen
2) Optimize the accommodation location based on list of activities chosen
3) Itinerary should follow specified number of days in the trip
4) Itinerary should follow specified number of hours in each day

The foundation of the problem formulation will be built on a network optimization problem where different nodes are connected to one another via arcs with multiple possible objectives derived from the network. The first step to formulating a network optimization problem is to plot its network diagram. *[Appendix 1]* For simplicity and legibility, only 5 activities were included in the network diagram.

The initial strategy was to solve the problem as a Traveling Salesman Problem (TSP). However, a TSP would be limited by a few restrictions such as its inflexible requirement to complete all activities, unable to optimize starting accommodation, unable to have multiple days and specify number of hours each day due to the requirement to complete all activities in 1 tour. Therefore, the Distance-Constrained Vehicle Routing Problem (DCVRP) has been identified as the main strategy to solve this problem. However, further modifications must be made to adapt the DCVRP to this problem.

Firstly, the DCVRP specifies that all activities must be completed. However, it may not be ideal to visit all places you wish to as the time required may be more than the amount of time you have, or it may result in a stressful and tightly packed itinerary. Therefore, the constraint that all activities must be visited has been removed.

Secondly, in the classic DCVRP formulation, there is only 1 depot node which means there is only 1 starting node. However, our model will have several depot nodes (accommodations) which the user has provided, and the model will be optimized to choose only 1.

Lastly, as the name suggests, instead of being distance-constrained, our model is travelling time and duration of each activity constrained as we should not spend more time than user specified each day. Therefore, the distance in the model is modified to the travelling time and the capacity in the original model is modified to be the total number of hours per day.

2.3 Overall Feasibility

The chosen modified DCVRP model is highly feasible because it can fulfil the 4 factors our project requires in a model to solve the business problem. The users will also receive a synthesized itinerary that is personalized based on user activities' preferences without them having to sacrifice working hours (Bansal, 2019).

3. Data Preparation

To illustrate a proof of concept, our model will be using datasets in the Singapore context.

3.1 Activities Dataset

An activities dataset has been prepared by searching up several activities an average tourist would like to do in Singapore. The dataset will contain 26 activities with details such as the addresses which will usually be given in their search for each activity, the estimated cost and duration they are expecting to spend at the activity and whether the activity is a "must-do". This will allow the model to be more precise and tailored to the users' needs. Moving forward, this excel could be also given to end users in the form of an online survey form where the user will be able to enter the above information intuitively to ensure data quality.

3.2 Accommodations Dataset

An accommodations dataset consisting of 6 different accommodations has been provided with details such as their addresses and per night prices for a standard room. Taking into consideration users with different income levels and priorities, the accommodations have been chosen from distinct locations around Singapore, all with a rating of 3 or above out of 5 stars and covers a wide price range between $100-$700.

3.3 Weather Conditions Dataset

A weather condition dataset consisting of the mapping of weather codes to weather conditions available from the WeatherAPI website. As the conditions are very specific, we have generalised and created a new “derived” column grouping similar weather conditions.

4. Model Formulation

4.1 Assumptions

The following assumptions are made when formulating the model:

- The model does not consider the restaurant preferences of a user and assumes the user is only prioritizing the itinerary of the activities and hence would eat at close by restaurants of their activities’ locations.
- All activities are assumed to be always available (24/7, ignoring public holidays)
- The user will stay at the same accommodation for the entire duration of the trip.
- The calculation of travelling time required between activities assumes the mode of transport is via car with an estimated average distance of 18.96 km per 31.82 minutes (~35.75 km/h) provided by Numbeo’s Traffic in Singapore article (Numbeo, 2023).

4.2 Network Optimization

Every activity node will be connected to and from every other node while every accommodation node is only connected to and from every activity node but not to other accommodation nodes. Therefore, the model is formulated as such (Figure 1):

Let $\mathcal{A} = \{0, 1, 2, \dots, N\}$ where N = No. of Accomodations + No. of Activities

Let $\mathcal{H} = \{0, 1, 2, \dots, M\}$ where M = No. of Accomodations and $\mathcal{H} \subset \mathcal{A}$.

Let u_i be a dummy variable for $i = 1, \dots, N$

Take $t_{ij} > 0$ to be the sum of the time taken to travel from activity i to activity j and the estimated duration required to be spent at j $\forall i \in \mathcal{A}, j \in \mathcal{A}$

Take R_i as a flag for activity i $\forall i \in \mathcal{A}$, 1 if user specified that it must be done this trip, 0 otherwise

Take C_j as the cost for activity j $\forall j \in \mathcal{A}$

Let K be the number of days of the trip and also number of subtours in the network optimization problem

Let D be the duration desired per day

Let T be the number of activities required per day, which is the maximum number of activities given by the performance of the model

Let B be the budget specified by the user of the trip, if none is specified the model will assign $1,000,000 by default which is a sufficiently high number that can cover all the cost of any trip

$$x_{ij} = \begin{cases} 1 & \text{the path goes from activity } i \text{ to activity } j, \quad \forall i \in \mathcal{A}, j \in \mathcal{A} \\ 0 & \text{otherwise} \end{cases}$$

$$h_i = \begin{cases} 1 & \text{accomodation } i \text{ has been selected,} \quad \forall i \in \mathcal{H} \\ 0 & \text{otherwise} \end{cases}$$

$$y_i = \begin{cases} 1 & \text{activity } i \text{ has been selected,} \quad \forall i \in \mathcal{A} \\ 0 & \text{otherwise} \end{cases}$$

$$\min \sum_{i \in \mathcal{A}} \sum_{j \in \mathcal{A}} t_{ij} x_{ij}$$

Figure 1: Model Formulation

The model has 3 decision variables: x to represent if any activity i to activity j is chosen, h to represent if the accommodation is chosen, and y to represent if the activity is chosen at all. A dummy variable, u, has been included as part of the Miller-Tucker-Zemlin (MTZ) Formulation.

The objective function is to minimize the total traveling time of all the activities and accommodation chosen.

$$
\begin{aligned}
&\text{s.t.} \sum_{i \in \mathcal{A}} x_{ij} \leq 1 && \forall j \in \mathcal{A} \setminus \{H\} && (1)\\
&\sum_{i \in \mathcal{A}} x_{ji} \leq 1 && \forall i \in \mathcal{A} \setminus \{H\} && (2)\\
&\sum_{j \in \mathcal{A}} x_{ij} = K h_i && \forall i \in \mathcal{H} && (3)\\
&\sum_{j \in \mathcal{A}} x_{ji} = K h_i && \forall i \in \mathcal{H} && (4)\\
&\sum_{i \in \mathcal{H}} h_i = 1 && && (5)\\
&\sum_{i \in \mathcal{A}} x_{ij} = x_{ji} && \forall j \in \mathcal{A} \setminus \{H\} && (6)\\
&u_j - u_i \geq t_{ij} - D(1 - x_{ij}) && \forall i, j \in \mathcal{A} \setminus \{H\}, i \neq j, d_{ij} \leq D && (7)\\
&0 \leq u_i \leq D && \forall i \in \mathcal{A} \setminus \{H\} && (8)\\
&\sum_{i \in \mathcal{A}} y_i \geq T && && (9)\\
&y_i \geq R_i && \forall i \in \mathcal{A} \setminus \{H\} && (10)\\
&y_j \leq \sum_{i \in \mathcal{A}} x_{ij} && \forall j \in \mathcal{A} && (11)\\
&\sum_{i \in \mathcal{A}} \sum_{j \in \mathcal{A}} x_{ij} C_j <= B && && (12)\\
&x_{ij} \in \{0, 1\} && \forall i, j \in \mathcal{A} && (13)\\
&h_i \in \{0, 1\} && \forall i \in \mathcal{H} && (14)\\
&y_i \in \{0, 1\} && \forall i \in \mathcal{A} && (15)\\
&x_{ii} = 0 && \forall i \in \mathcal{A} && (16)
\end{aligned}
$$

Figure 2: List of Constraints

From Figure 2, the list of constraints our model is subjected to are as follows:

- (1) and (2): To ensure that all activities are chosen at most once and the itinerary planned will not include repeated activities.
- (3) and (4): To ensure that the accommodation is visited exactly the number of times as the number of days of the trip. This means that the model will plan exactly the number of subtours for the number of days and the user will return to the hotel the exact number of times expected.
- (5): To ensure that only 1 accommodation is chosen throughout the trip.
- (6): The flow conservation constraint to ensure that when the user travels from activity i to activity j, the next activity must leave from the same activity j.
- (7) and (8): Constraints from the MTZ model to ensure that for each day in the itinerary, the combined travelling time and duration of activities planned do not exceed the number of hours within a day which is specified by the user's preferred travelling hours. It ensures that each time the user is done with activity i, the user still has enough time to move on to the next activity j.

- (9): To ensure that the number of activities chosen must be above a certain threshold T. In the absence of this constraint, minimising distance in the objective function would result in no activities selected at all as it will be the minimum distance covered. Therefore, a threshold ensures that as many activities can be selected to be fitted into the model. The selection of this threshold will be further explained in the next section.
- (10): To ensure that the user-specified "must-do" activities are satisfied, where specifying 1 if it is a "must-do" and 0 otherwise. This constraint will ensure that the activity is chosen if it is a "must-do" activity. Note: an illogical situation where all activities are flagged as "must-do" might cause the model to not return a solution at all.
- (11): To initiate variable y to ensure that y is 1 when the activity is chosen and 0 otherwise.
- (12): To ensure that the total cost of activities and accommodation chosen is within the budget set out by the user. The default budget is set at an arbitrarily high amount of $100,000 to ensure the model always satisfies this constraint if the user do not have a budget.
- (13), (14) & (15): To enforce $x, h\ and\ y$ variables to be binary.
- (16): To eliminate arcs from every activity/accommodation to itself in the model.

5. Outcome & Evaluation

5.1 Model Performance

The performance of our model is inversely proportional to the threshold T we select in Constraint (9), as a larger search space is required for higher numbers of locations, and hence a longer time to run the model. The time complexity of the model is exponential and represented in Big O notation as $O(2^n)$, as increasing the total number of activities and accommodations N by 1 would create an additional $(N-1)$ arcs in the network.

Ideally, the threshold T should be as large as possible to accommodate as many activities as possible. However, for our itinerary planner to be feasible and practical, the model must provide a solution within a reasonable timeframe according to each user. As such, we will use the user's specified time limit to help us determine the value of the threshold T.

Using the concept of binary search, we first set the low to $(number\ of\ days + 1\ for\ accommodation)$ (ensuring that there is at least one activity per day), and the high to $(number\ of\ activities + 1\ for\ accommodation)$. The midpoint of the low and high will be the first value of threshold T to be used in the model. After each run, we will cut the search space by half and repeat the process with different values of T, until the values of the low and high are equal. Finally, we can return the last optimal solution found within the time limit specified by the user.

```
Number of Activities:  15
Attempt no. 1
Set parameter Username
Academic license - for non-commercial use only - expires 2024-01-05
Being solved by Gurobi...
Solution status: 2
Running time: 0.3370s
Solution Found!
Number of Activities:  21
Attempt no. 2
Being solved by Gurobi...
Solution status: 2
Running time: 11.2360s
Solution Found!
Number of Activities:  24
Attempt no. 3
Being solved by Gurobi...
Solution status: 2
Running time: 90.8220s
Solution Found!
Number of Activities:  25
Attempt no. 4
Being solved by Gurobi...
Solution status: 9
Running time: 120.1470s
Unable to Find Solution!
```

Figure 3: Searching for Optimal Threshold T

5.2 Model Output

Using the following parameters below, we were able to create a 5-day itinerary with a detailed timeline (Figure 4):

- **Number of Activities**: 26
- **Number of Accommodations**: 6
- **Dates of the Trip**: 20 Apr 2023 – 24 Apr 2023
- **Start and End Time for each day**: 0900hrs - 2200hrs
- **Threshold T**: 24
- **Time Limit to Generate Solution**: 120 seconds
- **Budget**: Not specified (Default is $100,000)

```
This is your Singapore itinerary from 2023-04-20 to 2023-04-24

Start of  04/20/2023  from  Marina Bay Sands 9:00:00
Day 1 9:15 AM  -  11:15 AM : Fort Siloso
Day 1 11:19 AM  -  7:19 PM : Universal Studios Singapore
Day 1 7:31 PM  -  9:31 PM : National Gallery
End of  04/20/2023  at  Marina Bay Sands
Weather Forecast:
From 0900hrs to 1100hrs: Light Rain
From 1100hrs to 1200hrs: Heavy Rain
From 1200hrs to 1400hrs: Light Rain
From 1400hrs to 1500hrs: Moderate Rain
From 1400hrs to 2200hrs: No Rain
```

Figure 4: Snippet of Itinerary for First Day (Full Itinerary in Appendix 2)

```
In [14]: print("Minimised distance: ", "{:.1f}".format(results), "km")

Minimised distance:  60.4 km
```

Figure 5: Results of Minimised Distance

From Figure 5, the model was able to optimize the total distance of the entire itinerary to be 60.4 KM. In addition, we also used a weather forecast API (WeatherAPI) to predict the weather conditions for each of the days specified in the trip. Based on the weather forecast, the user can choose to rearrange the days in the itinerary without affecting the optimal distance.

Our itinerary can also be visualised by generating a map with a different layer/colour for each day showing the detailed route per day (Figure 6).

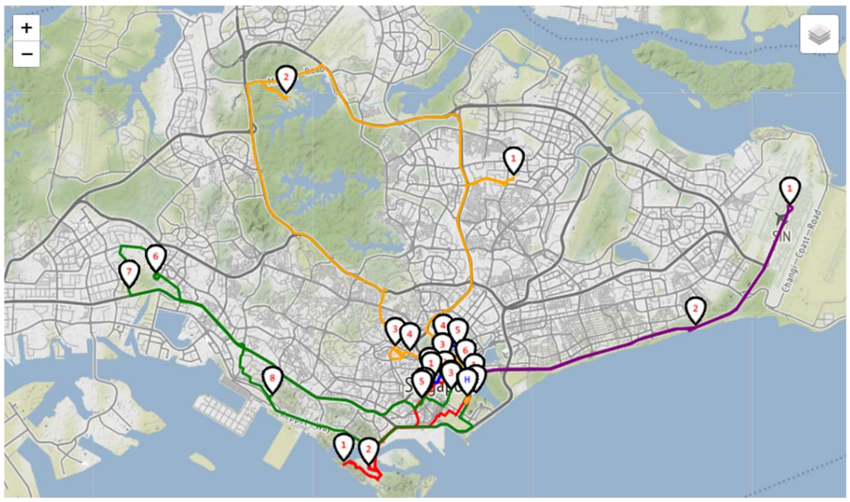


Figure 6: Route Map for Full Trip Itinerary

5.3 Model Limitations

While the outcome was as intended, there are still some flaws and limitations in the model which requires extensive work to refine it.

Firstly, the weather forecast is provided along with the output and not in the model's constraints which meant that outdoor activities can still be planned on days with inclement weather. However, weather conditions were excluded as a constraint because it could result in a lot lesser feasible solutions and there were also concerns with the forecast accuracy especially further into the future. A possible alternative solution could be to run another model using our current output to rearrange the days in the itinerary and minimise the number of outdoor activities planned in a time slot with inclement weather.

Secondly, the model did not consider mealtimes or recommend food options between activities. This is because we assumed that people would choose to eat at a place near their location and the travelling time would be negligible.

Lastly, the model did not consider the operating hours of the activities. This meant that an activity could be planned at a day or time when it is in fact not available. We excluded the operating hours as we did not find any free APIs that could generate the operating hours automatically based on the name of the activity, and it would be too troublesome for the user to input it manually. Adding the operating hours as constraints would also require us to track the starting time of each activity, making the model much more complex.

Therefore, while the model is a complete itinerary model built by basic optimization techniques, advanced techniques could be used to build on it and further enhance the model.

6. Future Work

6.1 Improving Usability with Automation and Paid APIs/Datasets

One major hurdle when formulating of the model was the lack of quality datasets and APIs. Several data sources and APIs were sourced to reduce the amount of effort by the end-user. However, like all open products, the inability to pay led to a limited number of calls with questionable activities. Web scraping from sites like Tripadvisor was also considered but scrapped due to potential legal issues. The inclusion of paid data sources could be used to generate more complete and clean lists of activities for users without preferences. Paid datasets such as accommodation datasets from STR, a data benchmarking company specializing in the hospitality industry includes cost and ratings of hotels. These detailed datasets will allow the end-user to have a more integrated accommodation selection experience and will minimize the amount of research and data required by the end-user.

A potential solution to improve usability is to build up a database and utilise machine learning and predictive analytics to predict the user's input while allowing for minor adjustments, instead of requiring raw data from the user. For example, we could allow a user to search for an activity and auto-populate the details of the activity, such as the estimated duration, attire, 'must-do' flag etc. based on historical data from other users. This can greatly reduce the effort required from the end user and make our itinerary planner more user-friendly.

6.2 Scalability

The scalability of the model is an important concern as our planner should ideally allow users to plan trips up to any length and any number of activities. While we have developed a quick fix to address the exponential running time of the model by introducing the threshold T and a time limit to run the model, it may not be ideal for larger problems (i.e., planning a month-long trip with only 25 activities).

One possible approach to solve this problem is to use heuristics and find ways to minimise the distance in a faster and more efficient manner. The results may not be optimal, but it can still provide multiple benefits to the user as the optimality of the distance is not as crucial as allowing for more activities to be planned.

Another approach to improve the scalability could be to subdivide the problem into smaller problems. For example, a month-long trip with 100 activities can be subdivided into 4 week-long trips with 25 activities each. We can plot all the activities using their latitude and longitude on a graph and use clustering techniques to group them, minimising the intra-cluster distances. From there, we can run our model on each sub-group and combine the results to create a month-long itinerary.

6.3 Inclusion of Additional Constraints

With our current model, it is possible to extend our constraints to include attire (i.e., ensuring activities within the same day require the same attire) or even the type of activities allowed (i.e., no back-to-back physical activities). This can allow for more user customisability and cater to more specified user needs.

To develop an even more sophisticated and detailed model, we could modify our decision variables to include the start and end time of each activity. From there, our team will be able to implement additional constraints such as the consideration of meals and operating hours, which existing itinerary planners on the market are not considering.

For example, we could now include food activities as nodes into the network and restrict all days with time slots between 8AM-10AM, 12PM-2PM and 6PM-8PM to include a meal activity and all other time slots to not include any meal activities. These meals could be a type of recommendation engine as part of the minimizing distance model. The list of restaurants with its ratings could be included as a subtype of activities and more constraints could be added such as a minimum required rating or ensuring different cuisines in a day.

7. Conclusion

In conclusion, the model has answered our problem statement and was proven to be highly feasible, exhibiting vast potential to be expanded upon and is innovatively comparable to existing solutions. Although given the lack of advanced modelling techniques and budget set out for this project, the model is still lacking certain constraints that would have made it more all-encompassing and customisable for the end-user’s needs. Nonetheless, our strategy was a success reducing planning time significantly from days/weeks to just a few minutes.

8. References

Bansal, A. (2019, December 11). The Pain-Points and Opportunities in Travel - XOKind - Medium. Medium. https://medium.com/xokind/the-opportunities-and-pain-points-in-travel-d2c5cc06ae1a

Chen, Z., Sim, M., & Xiong, P. (2021). RSOME in Python by XiongPengNUS. https://xiongpengnus.github.io/rsome/about

Ekstein, N. (2017, October 25). Why You Should Usually Book Your Hotel a Month Before Your Flight. Bloomberg.com. https://www.bloomberg.com/news/articles/2017-10-25/why-you-should-usually-book-your-hotel-a-month-before-your-flight#xj4y7vzkg

GeoPy 2.3.0 documentation. (n.d.). https://geopy.readthedocs.io/en/stable/#nominatim

Nawijn, J., Marchand, M. a. G., Veenhoven, R., & Vingerhoets, A. J. J. M. (2010). Vacationers Happier, but Most not Happier After a Holiday. Applied Research in Quality of Life, 5(1), 35–47. https://doi.org/10.1007/s11482-009-9091-9

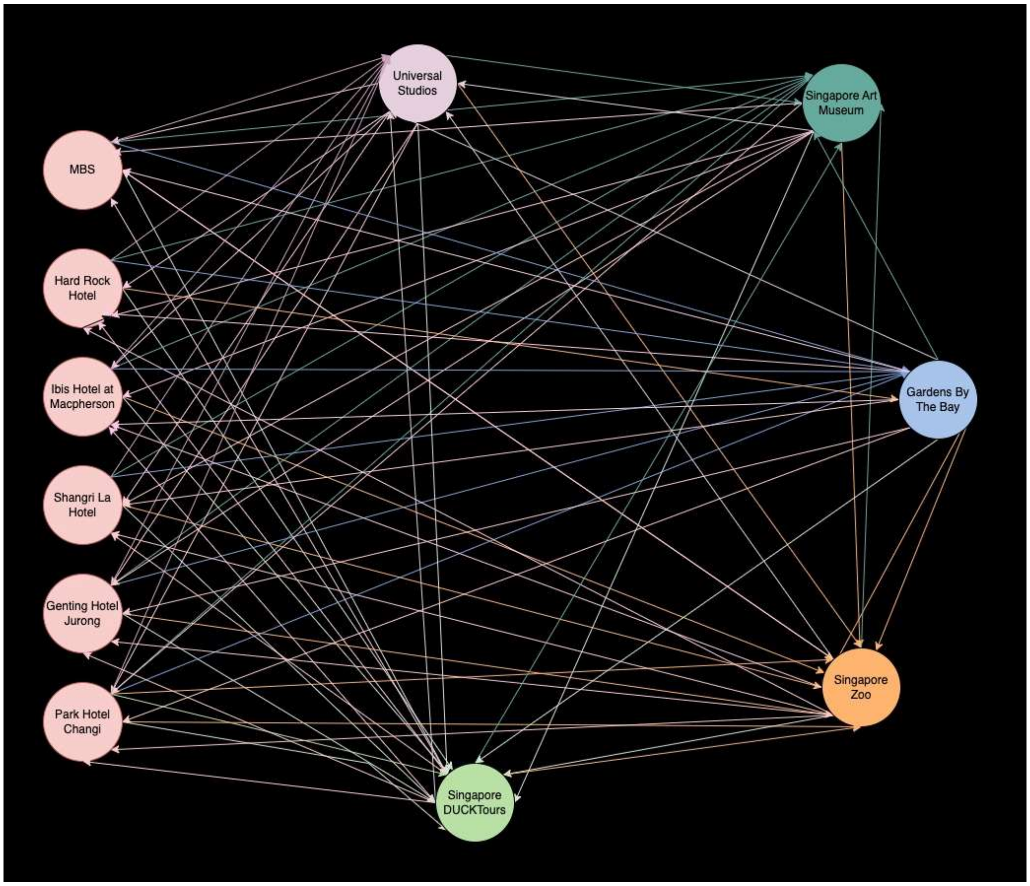
Openrouteservice API. (n.d.). https://openrouteservice.org/

OpenWeatherMap.org. (n.d.). Weather API - OpenWeatherMap. https://openweathermap.org/api

Russo, C. H. (2016, June 8). The Happiest Part Of Your Vacation Isn’t What You Think. HuffPost. https://www.huffpost.com/entry/how-to-make-the-most-of-vacation_n_5755b42ae4b0eb20fa0e906d#:~:text=It's%20perhaps%20not%20surprising%20then,escapism%20to%20their%20daily%20routines.&text=Of%20course%2C%20you%20should%20actually%20take%20the%20vacation%20too.

Traffic in Singapore. (n.d.). https://www.numbeo.com/traffic/country_result.jsp?country=Singapore

9. Appendices



Appendix 1: Extract of Network Diagram of Model

```
Start of  04/20/2023  from  Marina Bay Sands 9:00:00
Day 1 9:15 AM  -  11:15 AM : Fort Siloso
Day 1 11:19 AM  -  7:19 PM : Universal Studios Singapore
Day 1 7:31 PM  -  9:31 PM : National Gallery
End of  04/20/2023  at  Marina Bay Sands
Weather Forecast:
From 0900hrs to 1100hrs: Light Rain
From 1100hrs to 1200hrs: Heavy Rain
From 1200hrs to 1400hrs: Light Rain
From 1400hrs to 1500hrs: Moderate Rain
From 1400hrs to 2200hrs: No Rain


Start of  04/21/2023  from  Marina Bay Sands 9:00:00
Day 2 9:05 AM  -  11:05 AM : Clarke Quay
Day 2 11:05 AM  -  12:05 PM : Fort Canning
Day 2 12:08 PM  -  3:08 PM : Singapore Art Museum
Day 2 3:10 PM  -  5:10 PM : Little India
Day 2 5:11 PM  -  7:11 PM : Arab Street
Day 2 7:15 PM  -  9:15 PM : Singapore DUCKtours
End of  04/21/2023  at  Marina Bay Sands
Weather Forecast:
From 0900hrs to 1000hrs: Light Rain
From 1000hrs to 1100hrs: Heavy Rain
From 1100hrs to 1500hrs: Light Rain
From 1500hrs to 1600hrs: No Rain
From 1600hrs to 1700hrs: Light Rain
From 1700hrs to 1800hrs: No Rain
From 1800hrs to 1900hrs: Light Rain
From 1900hrs to 2000hrs: No Rain
From 1900hrs to 2200hrs: Light Rain

 Start of  04/22/2023  from  Marina Bay Sands 9:00:00
 Day 3 9:02 AM  -  10:02 AM : Singapore Flyer
 Day 3 10:05 AM  -  10:35 AM : Esplanade
 Day 3 10:35 AM  -  11:05 AM : Merlion
 Day 3 11:08 AM  -  1:08 PM : Chinatown
 Day 3 1:08 PM  -  2:08 PM : Keong Saik Rd
 Day 3 2:34 PM  -  6:34 PM : Singapore Science Centre
 Day 3 6:43 PM  -  7:43 PM : Jurong Lake Gardens
 Day 3 7:59 PM  -  8:59 PM : Haw Par Villa
 End of  04/22/2023  at  Marina Bay Sands
 Weather Forecast:
 From 0900hrs to 1000hrs: Heavy Rain
 From 1000hrs to 1100hrs: Light Rain
 From 1100hrs to 1500hrs: Heavy Rain
 From 1500hrs to 1600hrs: Light Rain
 From 1600hrs to 1700hrs: Heavy Rain
 From 1700hrs to 2100hrs: Light Rain
 From 2100hrs to 2200hrs: Heavy Rain
```

```
Start of  04/23/2023  from  Marina Bay Sands 9:00:00
Day 4 9:32 AM  -  11:32 AM : Jewel
Day 4 11:45 AM  -  2:45 PM : East Coast Park
Day 4 3:04 PM  -  7:04 PM : Gardens By The Bay
End of  04/23/2023  at  Marina Bay Sands
Weather Forecast:
From 0900hrs to 1000hrs: Drizzle
From 1000hrs to 1100hrs: Heavy Rain
From 1100hrs to 1200hrs: Drizzle
From 1200hrs to 1300hrs: Light Rain
From 1300hrs to 1400hrs: Drizzle
From 1400hrs to 1500hrs: Light Rain
From 1500hrs to 1600hrs: No Rain
From 1600hrs to 1700hrs: Light Rain
From 1600hrs to 2200hrs: No Rain


Start of  04/24/2023  from  Marina Bay Sands 9:00:00
Day 5 9:24 AM  -  9:54 AM : Kampong Lorong Buangkok
Day 5 10:23 AM  -  6:23 PM : Singapore Zoo
Day 5 6:56 PM  -  8:56 PM : ION Orchard
Day 5 8:57 PM  -  9:57 PM : Istana
End of  04/24/2023  at  Marina Bay Sands
Weather Forecast:
From 0900hrs to 1000hrs: Light Rain
From 1000hrs to 1100hrs: No Rain
From 1000hrs to 2200hrs: Light Rain
```

Appendix 2: Full 5-day Itinerary