**IS602 Spreadsheet Modelling for Decision Making**

**Final Report: Group 5**

Prepared by: Chiu Shih-Cheng, Fong Bao Xian, Loh Jiahui, Neo Yi Xin, Shi Kaiyue, Zhu Yiting

Date: 31 October 2022

# Introduction

Dawes and Rowley (1996) described waiting time as a common experience, and part and parcel of any leisure experience. In a study conducted by Li (2010), it was highlighted that waiting time, specifically perceived waiting time, waiting information and waiting environment had significant impact on theme park satisfaction among tourists. The issue of crowding and long waiting times also tended to be more severe for theme park visitors due to uneven distribution within the park (Brown et. al, 2013).

This project aims to build a customer-centric model that will maximise users’ experience at the Universal Studios Singapore (USS). The model is constructed to take in users’ preferences such as their time of visit and preferred rides and return an optimal schedule that will minimise their total wait time and travel time between rides. To cater for different planning styles, the model also allows users to choose between a high-level hourly schedule and a more detailed plan down to the minute.

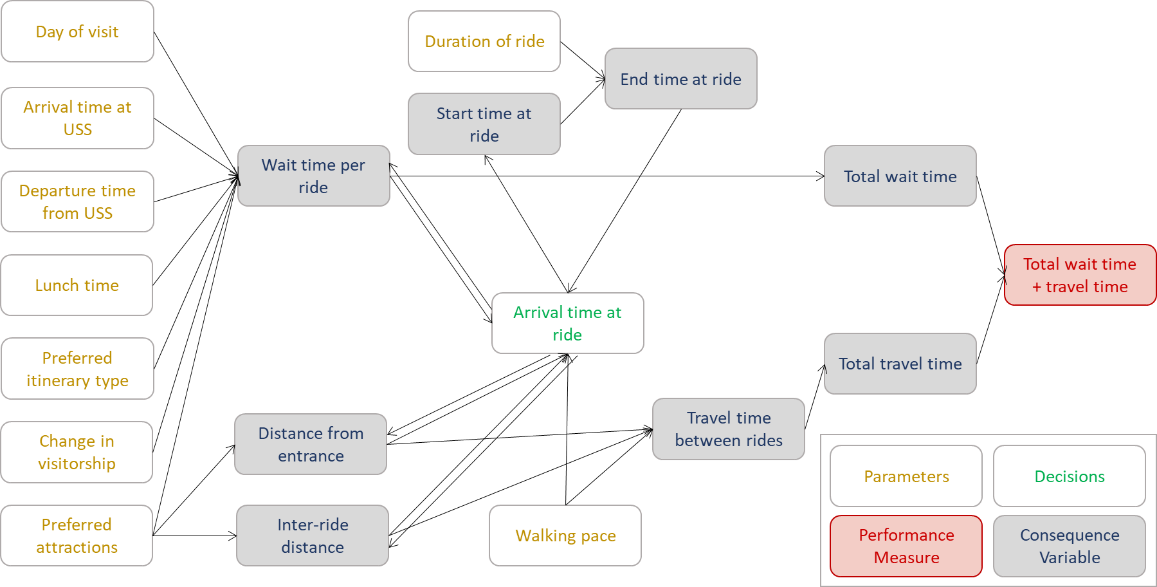
# Data Sources

The following data sources are used:

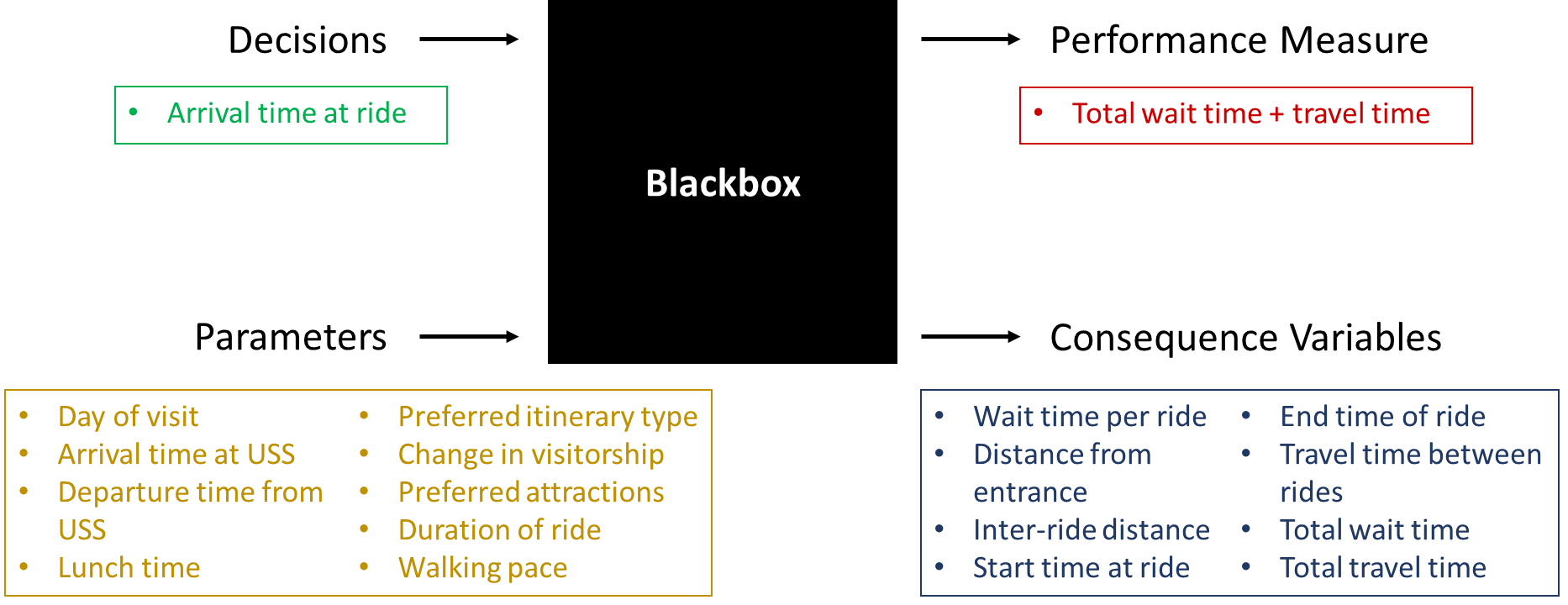
1. **Universal Studios Singapore App:** Real-time wait times for attractions
2. **Prepare Travel Plans website (2022):** Duration of rides
3. **Google Maps:** Estimated distance between attractions
4. **Research study by Browning et al. (2006):** Average walking pace

# Influence Diagram and Blackbox Model

## Influence Diagram



## Blackbox Model



# Model Design and Building

## User Interface

At the User Interface (UI), the user is first asked to select whether a **flexible (high-level) plan** or **back-to-back (detailed) plan** is preferred. Each plan is computed by a different model, which will be explained in Section 4.3 and 4.4 respectively.

Selection of plan will bring the user to the relevant spreadsheet where user will be asked to input their choices on the following parameters:

1. **Expected day of visit** (Wednesday to Sunday – Park is closed on Monday and Tuesday)
2. **Expected time of arrival and departure** (Between 11am and 6pm)
3. **Lunch preference** (Yes/No)
4. **Desired lunch time and duration** (1 or 2 hours)
5. **Up to 5 prioritised rides** (Dropdown list of all USS rides)
6. **Preferred pace of travel** (Average or Leisurely)

Upon entering all inputs and clicking a button to proceed, the chosen model is triggered, and the user will be brought to the schedule page (refer screenshot below) with the recommended itinerary that has the shortest total wait and travel time. A map view of the route is also displayed to guide users to their destinations. Macros are used to automate the triggering of models and population of results from the model to the schedule page.

A picture containing table

Description automatically generated

## Data Collection, Simulation and Computation

As a base for computation in our models, the following reference data are collected and generated:

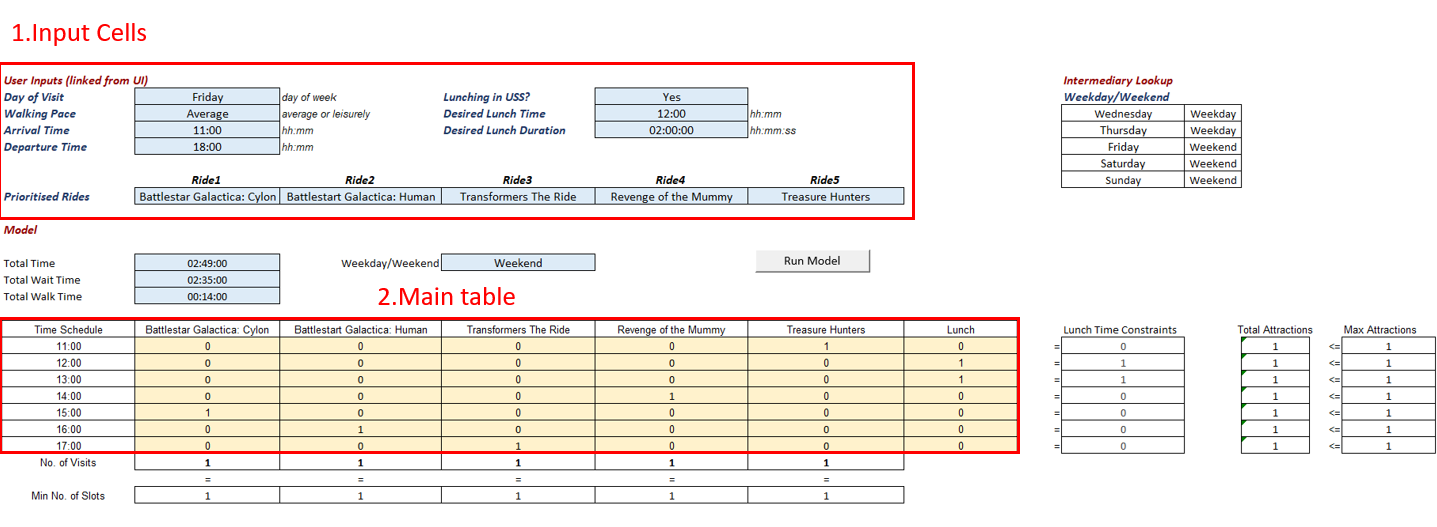
|  |  |  |
| --- | --- | --- |
| **S/N** | **Description** | **Formula** |
| 1. **Master Wait Times worksheet (Weekday / Weekend)** | | |
| 1 | Real-time ride wait times are collected from the USS App on an hourly basis[[1]](#footnote-2). As only a limited set of actual wait times could be collected due to time constraints, resampling is conducted to simulate 100 additional representative sample wait times for each ride at each hour to support our model. | =PERCENTILE(*raw\_ data*,RAND()) |
| 2 | A single average wait time for each ride and hour is computed from the 100 samples. | =AVERAGE(*100\_ samples*) |
| 3 | The average wait times are rounded up to the nearest 5 minutes to (i) more closely model the wait times generated by the USS App which are in steps of 5-minutes and (ii) allow for some buffer time. | =CEILING(*average\_ time*,TIME(0,5,0)) |
| 4 | S/N 1-3 are repeated for both Weekdays and Weekends[[2]](#footnote-3) to account for the different wait time patterns. The resulting wait times are tabulated in two separate tables – **Weekday Wait Time** and **Weekend Wait Time**. | |
| 1. **Master Ride Times worksheet** | | |
| 5 | Ride durations are taken from the travel blog, Prepare Travel Plans[[3]](#footnote-4). To allow for margins of error, all ride durations are rounded up to the nearest minute. All ride durations are tabulated in the **Ride Time** table. | =CEILING(*ride\_ time*,TIME(0,1,0)) |
| 1. **Master Walk Times worksheet (Average / Leisurely)** | | |
| 6 | Walk times between different rides and with the entrance are computed using (i) estimated distance from Google Maps and (ii) estimated average walking speed of 1.4 metres per second based on literature, rounded up to the nearest minute. | =CEILING(TIME(0,0, *distance*/*walking\_ speed*),(TIME(0,1,0)) |
| 7 | The table of walk time computation is replicated with a slower walking speed at 0.7 metre per second (50% of average walking speed). This is considering some visitors may be travelling with young children and/or elderly and wish to take a more leisurely pace. The resulting walk times are tabulated in two separate tables – **Average Walk Time** and **Leisurely Walk Time**. | |

## Model 1 – High-Level Plan (Solver Method)

The first model uses Excel Solver twice to find the best combination of rides and time slots that will minimise total wait and walk times of the user. The main computations used in this model are explained in the following:

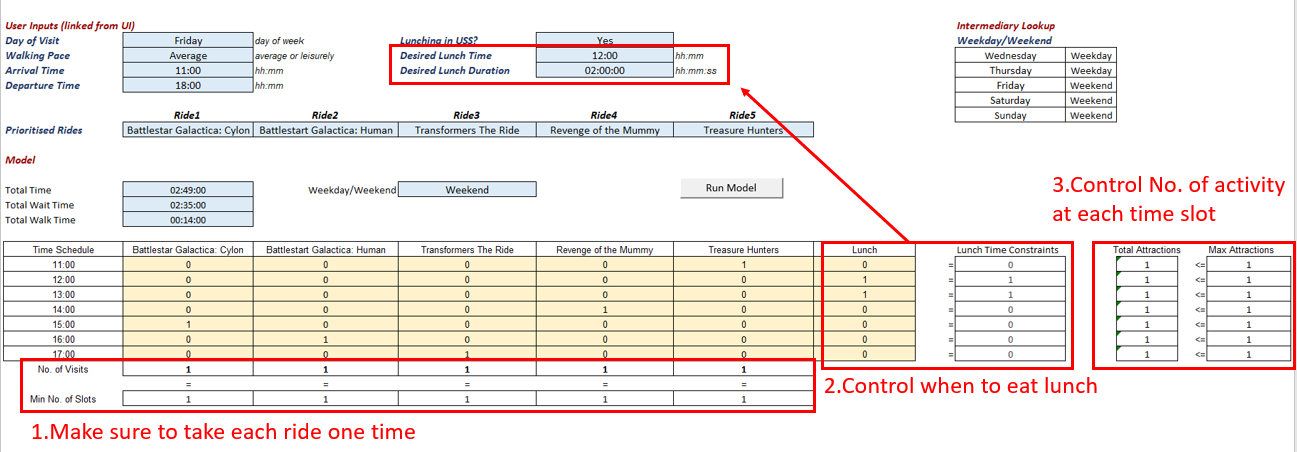
Input Cells and Main Table

1. Input cells make reference to what the user selects in the UI page.
2. The schedule is split into hourly time slots (rows) which are blocked out for 1 activity (column) each. This means for the full hour, the user may be advised to take 1 ride or to have lunch. The cells in the Main Table will be our variable cells that we want Solver to change. These cells will take on binary values, with “*1*” representing what the user will do in that time slot.



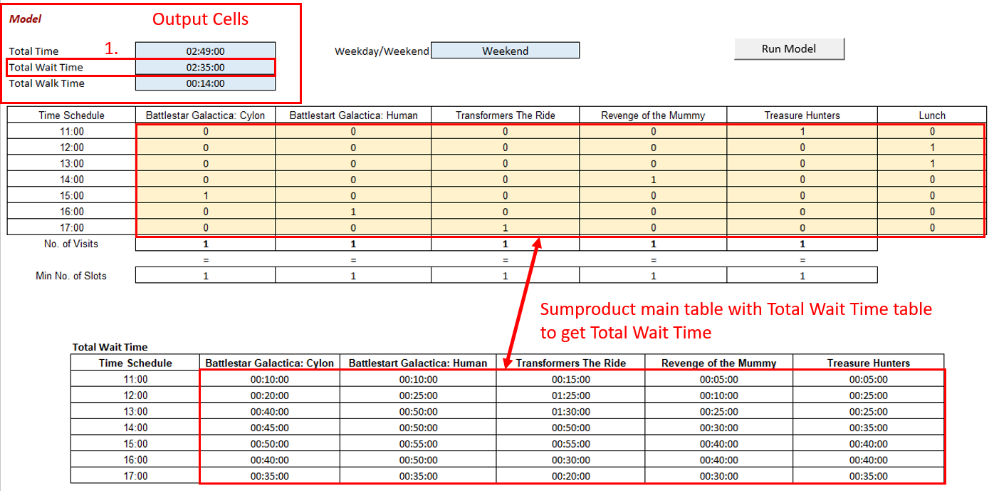
Constraints

1. *No. of Visits* sums up how many times each ride is taken. This constraint is set to ensure each ride chosen by the user is taken exactly once.
2. *Lunch Time Constraints* is set based on whether the user wants to have lunch, the desired lunch time and duration. In this example, the user wants to have lunch at 12:00 and for 2 hours, so the rows for the 12:00 and 13:00 are set to “*1*”. Later in the Solver, we will set a constraint for *Lunch Column* to be equal to *Lunch Time Constraints*, so we can make sure that 12:00 and 13:00 are reserved for lunch and not used for rides.
3. *Max Attractions* is set such that the sum of each row in the main table (*Total Attractions*) is ≤ 1 activity.



Objective Cells (Output Cells)

1. *Total Wait Time*: Using SUMPRODUCT() on the Main Table with Total Wait Time table, we can get the *Total Wait Time*. For example, at 11:00, only 1\*00:05:00 will return a non-zero value, so total wait time for 11:00 will be 5 minutes. SUMPRODUCT() sums up these wait times for each time slot to arrive at *Total Wait Time* for the entire visit.

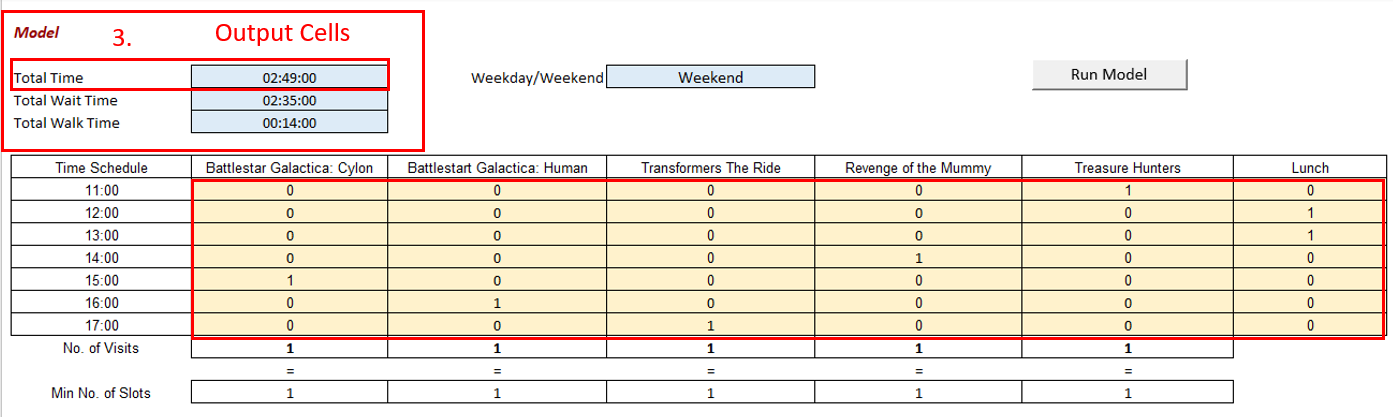


1. *Total Walk Time*: Based on the binary values in the Main Table, the location of the user in the schedule is mapped to the Location Mapping Table. The walk times between locations are then fetched from the **Average** or **Leisurely Walk Time** table using INDEX(MATCH( MATCH())) depending on the user’s preferred pace. In the below example, to reach the Treasure Hunters ride at the 11:00 slot, the user needs to walk from the entrance in the row above to Treasure Hunters in the current row, which is estimated to take 6 minutes at average pace. At 12:00, because we assume the user will have lunch at his or her present location (Treasure Hunters), the walk time is 0 minute. The walk time column is summed up to get the *Total Walk Time*.

A picture containing chart

Description automatically generated

1. *Total Time*: This is the sum of *Total Wait Time* and *Total Walk Time* and is the objective cell that we want to minimise in our Solver.



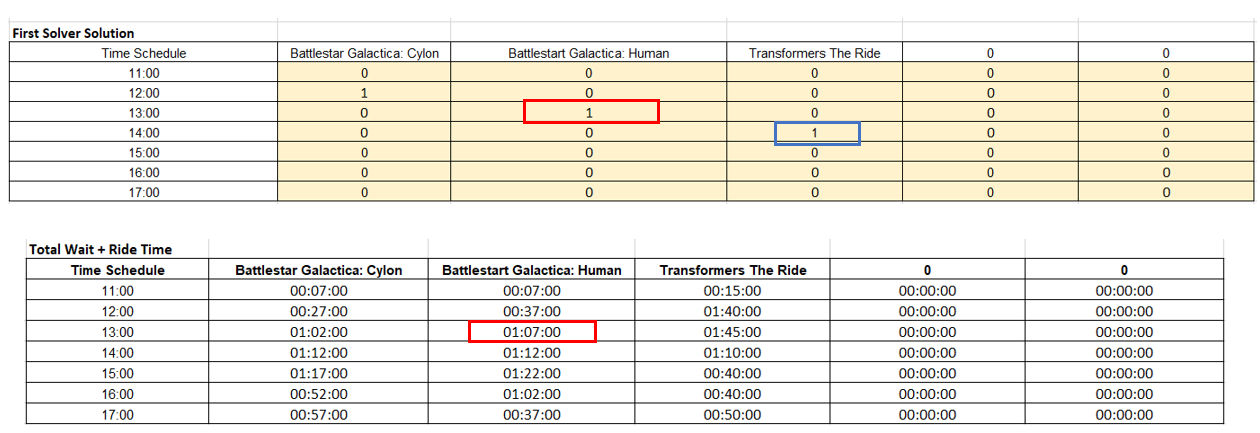
Solver Settings (First Solver)

1. Our first Solver will be set to minimise *Total Time* by changing the binary cells in the Main Table, subjected to the constraints explained above.

Graphical user interface, application

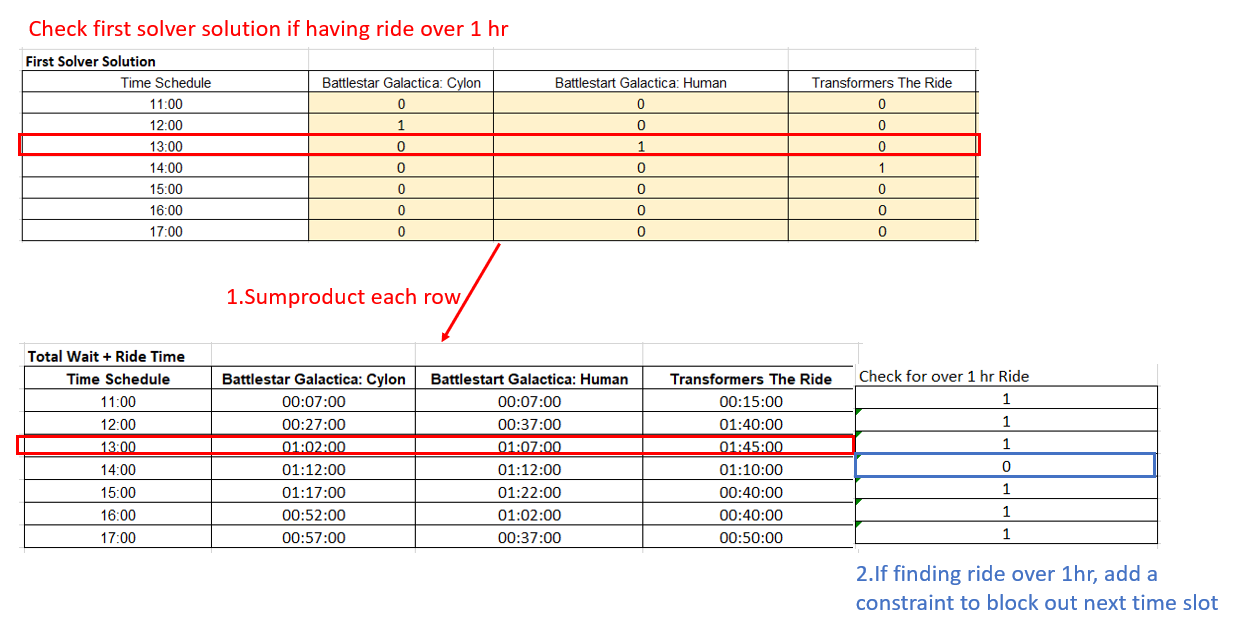
Description automatically generated

1. However, the outcome of this Solver does not account for wait times that exceed 1 hour. In the example below, at 13:00, the combined wait time and ride time of Battlestar Galactica: Human is 1 hour and 7 minutes, which exceeds the 1-hour time slot. Thus, we should not have any rides at 14:00. To account for this, we need to add one more constraint and use Solver a second time.

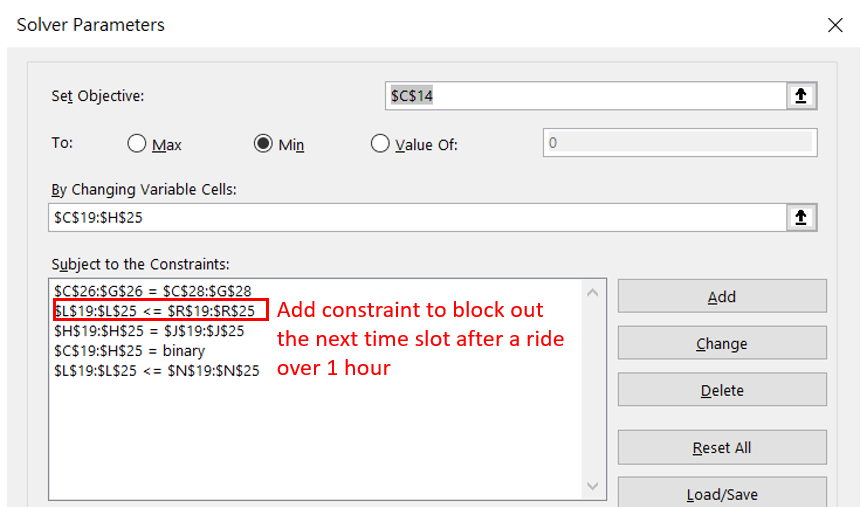


Solver Settings (Second Solver)

1. To check for rides exceeding 1 hour, results from the Main Table generated from the first Solver are copied to a different location. Then, SUMPRODUCT() is computed on each row to check whether any of the rides and slots allocated a “*1*” in the first Solver exceeds 1 hour. If there is, in a new column (*Check for over 1hr Ride*), 0 is set to indicate that the next time slot should not be assigned.



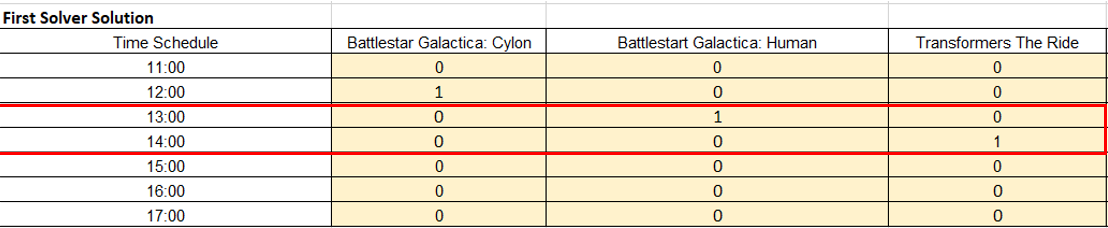
1. This new constraint is added into Solver, with all other settings remaining the same. We then run Solver a second time to get the final results.



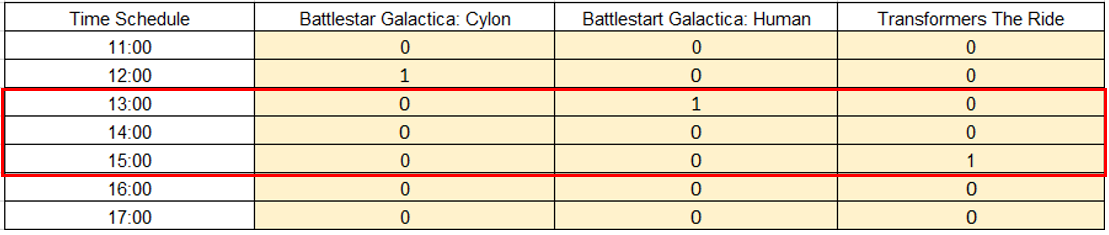
Comparison of results from First Solver and Second Solver

1. In the First Solver solution, 14:00 is not blocked out and is used for Transformers The Ride, even though the Battlestar Galactica: Human wait and ride time exceeds 1 hours at 13:00. In the Second Solver solution, the 14:00 time slot is blocked out accordingly and Transformers The Ride is shifted from 14:00 to 15:00.

*First Solver solution*



*Second Solver solution*



## Model 2 – Detailed Plan (Permutation Method)

The second model generates all permutations of the itinerary based on user inputs (3! = 6, 4! = 24, 5! = 120 permutations for 3, 4 and 5 rides respectively), mainly using an INDEX(MATCH(MATCH())) approach. Depending on the number of attractions selected, a different permutation table for 3, 4 and 5 rides is populated. An example of the permutation table (first 10 rows out of 120 permutations) and the key formulae are shown below:

Table

Description automatically generated with medium confidence

Table

Description automatically generated Table

Description automatically generated

Repeat “Lunch?” to “Ride 1 End time” columns 4 times for Rides 2-5

| **S/N** | **Column name** | **General formula** |
| --- | --- | --- |
| 1 | USS arrival time | Reference user input cell |
| 2 | Lunch? | =IF(*lunch\_input*=”No”,0,IF(*arr\_time*>=*preferred\_lunch\_time*,1,0)) |
| 3 | Entrance to ride 1 | =IF(*walking\_speed*="Average",  INDEX(*master\_walk\_times\_average*,  MATCH(*entrance*,*master\_walk\_times\_average\_col\_ref*,0),  MATCH(*ride1*,*master\_walk\_times\_average\_row\_ref*,0)),  INDEX(*master\_walk\_times\_leisure*, MATCH(*entrance*,*master\_walk\_times\_leisure\_col\_ref*,0),  MATCH(*ride1*,*master\_walk\_times\_leisure\_row\_ref*,0))) |
| 4 | Arrival time at ride 1 | =IF((*arr\_time*+*lunch\_tag*\**lunch\_duration*+*entrance\_to\_ride1*)> *dep\_time*,  "",*arr\_time*+*lunch\_tag*\**lunch\_duration*+*entrance\_to\_ride1*) |
| 5 | Wait time at ride 1 | =IF(*arr\_time\_ride1*="","",  IF(*preferred\_day*="Weekday",  INDEX(*master\_wait\_times\_weekday*,  MATCH(*arr\_time\_ride1*,*master\_wait\_times\_weekday\_col\_ref*,1),  MATCH(*ride1*, *master\_wait\_times\_weekday\_row\_ref*,0)),  INDEX(*master\_wait\_times\_weekend*,  MATCH(*arr\_time\_ride1*,*master\_wait\_times\_weekend\_col\_ref*,1),  MATCH(*ride1*, *master\_wait\_times\_weekend\_row\_ref*,0)))) |
| 6 | Play time at ride 1 | =IF(*wait\_time\_ride1*="","",  VLOOKUP(*ride1*,*master\_ride\_times*,2,FALSE)) |
| 7 | Ride 1 end time | =IF(*play\_time\_ride1*="","",  *arr\_time\_ride1*+*wait\_time\_ride1*+*play\_time\_ride1*) |
| 8 | Total walk time | =IF(*ride5\_end\_time*="",*error\_msg*,  IFERROR(*walk\_time\_ride1*+*walk\_time\_ride2*+*walk\_time\_ride3*+ *walk\_time\_ride4*+*walk\_time\_ride5*,*error\_msg*)) |
| 9 | Total wait time | =IFERROR(*wait\_time\_ride1*+*wait\_time\_ride2*+*wait\_time\_ride3*+ *wait\_time\_ride4*+ *wait\_time\_ride5*,*error\_msg*) |
| 10 | Total (walk + wait) time | =IFERROR(SUM(*total\_walk\_time*+ *total\_wait\_time*),*error\_msg*) |

Thereafter, the minimum total walk + wait time is found based on the MIN() function, and the FILTER() function is used to return all permutations which gives this minimum time. The first permutation is returned as output to the user as the recommended detailed schedule in the schedule page. Should the user wish to explore other options, the full list is available in the Model 2 worksheet.

# Trade-off Analysis

The following trade-off cases have been considered in our model. Depending on the user’s choices of inputs, the model will consider them in computing the optimal itinerary that will minimise total wait and walk times.

## Day of Visit

Depending on the user’s expected day of visit, the model will take in inputs from the **Weekday Wait Time** or **Weekend Wait Time** table accordingly. Based on the wait times collected, total wait times were at certain timings higher during the weekdays compared to the weekends. While the team is uncertain of the reason behind this trend, we postulate that this may be due to more visitors preferring to visit during the weekdays, thinking that it would be less crowded or there is less manpower assigned to manage the park during the weekdays leading to slower queues.

## Time of Arrival and Departure

The user’s total duration in the park will affect the number of rides that he or she could take. The time of arrival also turned out to be a significant factor that impacts overall wait times. Specifically, wait times tend to be much shorter in the first hour of park opening, and peak in the mid-afternoon. As such, arriving early will enable clearing more rides or the popular rides in the early hours and optimise total wait times.

## Lunch Time and Duration

Should a user choose to have lunch in USS, the model will allocate the lunch slots accordingly and this will affect the optimal itinerary. It will be useful to consider ride wait time trends when planning lunch. For example, deciding to have lunch at 11am (opening hour) will mean that the user will not benefit from the shortened wait times across multiple attractions during this period.

## Preferred Pace of Travel

Depending on the user’s preferred pace of travel, the model will draw inputs from the **Average Walk Time** or **Leisurely Walk Time** table accordingly. A slower pace will lengthen total walk time and may affect the optimal itinerary because different arrival time at a particular ride will have different wait times.

# Sensitivity Analysis

For our sensitivity analysis, we considered the scenario where wait times (for both weekdays and weekends) were to change. As travel restrictions lift globally, all attractions, including the USS, should expect visitor numbers to slowly increase and return to pre-pandemic numbers. As visitor numbers are directly correlated with ride wait times, this may increase total wait times. On the contrary, if there were to be a rebound of cases or other factors leading to tightening of safe management measures or travel restrictions, visitor numbers and hence wait times will decrease.

Our sensitivity analysis will be performed on a fixed case of user input selections as follows:

* *Expected time in USS*: 12pm to 6pm
* *Lunch Time and Duration*: 1pm, for 1 hour
* *Prioritised attractions*: (i) Battlestar Galactica: Cylon, (ii) Battlestar Galactica: Human, (iii) Transformers The Ride, (iv) Revenge of The Mummy and (v) Treasure Hunters

For Model 1, total wait times increase from around 2 hours to 4 hours with increase in visitor numbers from -30% to +30% of the current level. From the graphs, Model 1 is observed to be less sensitive and hence more robust to visitor volume changes on weekends, as can be seen from the smaller change in total wait times. This is consistent with our observation that the average wait times tend to be shorter on weekends for most attractions.

By contrast, Model 2 is less sensitive and hence more robust to visitor volume changes on weekdays, with a smaller gap between total wait times in most scenarios. However, the results are less uniform for Model 2, especially between 10% and 20% increase of the visitor numbers, perhaps due to a popular ride being pushed to the next hour resulting in a sharp increase in the total wait time.

Chart

Description automatically generated Chart, line chart

Description automatically generated

# Model Limitations

Our current model has the following limitations:

## Limitations of 1-hour slots in Model 1 (High-Level Plan)

Model 1 blocks entire time slots by the hour, even if the wait and ride time is only 10 minutes, and hence may not fully maximise the user’s time in USS. Nevertheless, this is intended for users who prefer greater flexibility, and it is sufficient for them to know that they should reach the next attraction by the start of the next hour, so they can spend the rest of their time on other activities like shopping or watching parades.

## Less room for variation in Model 2 (Detailed Plan)

On the other hand, Model 2 provides a schedule that is packed back-to-back, and provides less room for variation like shopping etc. This is more suitable for users who wish to maximise their time at USS to complete their attractions of choice upfront, and then take the remaining time to complete other activities. In fact, the users can repopulate a second schedule with their next 5 attractions of choice after completing their first 5 using Model 2. Even if they were to miss certain timings, they can simply regenerate the latest recommended schedule.

## Limitations of collected wait times

As wait times were collected on an hourly basis, our model has made assumptions that wait time for the ride anytime within the hour is the same. We have also not considered how rain will impact our models due to insufficient data collected. Future enhancements may consider collecting wait times at higher frequencies or for longer period of time and over different weather conditions to see how that may affect the model.

## Parade timings not taken into consideration

Our model does not consider parade timings as they are spread around the park at different times of the day. Users wishing to watch parades have to incorporate such plans into their schedule separately.

# References

Brown, A., Kappes, J., & Marks, J. (2013). *M*itigating Theme Park Crowding with Incentives and Information on Mobile Devices. *Journal of Travel Research, 52*(4), 426–436.

Browning, R. C., Baker, E. A., Herron, J. A., & Kram, R. (2006). Effects of obesity and sex on the energetic cost and preferred speed of walking. *Journal of Applied Physiology, 100*, 390–398.

Dawes, J., & Rowley, J. (1996). The waiting experience: towards service quality in the leisure industry. *International Journal of Contemporary Hospitality Management, 8*(1), 16–21.

Li, W. L. (2010). Impact of waiting time on tourists satisfaction in a theme park: An empirical investigation. *2010 IEEE International Conference on Industrial Engineering and Engineering Management.*

Universal Studios Singapore: How to Go & 15 Best Rides. (2022, September 11). In *Prepare Travel Plans.* https://preparetravelplans.com/universal-studios-singapore-rides-guide/

1. During this project, USS’ opening days and hours are from Wednesday to Sunday, 11am to 6pm. Wait times were collected hourly across 13 days – between 30 Sep and 16 Oct 2022, 11am to 5pm. Due to weather conditions impacting wait times and updates on the USS App, 2 days of wait time data were eventually excluded. [↑](#footnote-ref-2)
2. Actual wait times for Fridays are noted to be more similar to Saturdays and Sundays than the other weekdays. Hence, for the purpose of this project, Wednesday and Thursday will reference weekday timings, while Friday to Sunday will reference weekend timings [↑](#footnote-ref-3)
3. As the ride duration for the Magic Potion Spin is not available, a 5-minute ride duration is assumed based on the team’s past personal experience. [↑](#footnote-ref-4)