ANLY482 ANALYTICS PRACTICUM

Car Park Overspill Study

Final Report

Team BLACKBOX:

CHEN XUEYE

ZENG JIADONG

ZHENG WEI

Contents

1. Abstract

Together with the constant development of our society, density of urban populations have increased and cities' infrastructure have become increasingly complex. In the aspect of transportation, one of the crucial issue is the management of urban traffic growth. Singapore being a small country, has limited land space. Hence, in order to utilize the land and improve the city infrastructure, Land Transport Authority(LTA) has conducted a Singapore Parking Situation Study which aims to understand the current parking situation in Singapore at selected locations.

This report explains how we built an Agent-based Modelling tool in NetLogo by incorporating the data collected in LTA's study to simulate Singapore public car park traffic situation in order to help Singapore car park designers and engineers to visualize possible traffic outcomes against different car park design factors. The report covers introduction of this project, literature review that helps define our project methodology, the project methodology we used in building our model, analysis and evaluation of our model results, and areas that we have identified to further improve our model.

2. Introduction

Being interested to know analytics problems underlining the management of Singapore car parks, our group proposed this project aiming to build a simulation model to allow car park designers and engineers to try a series of different parameters in the model and to simulate the traffic conditions in the car park. In order to build the model, we need the realistic data showing the pattern of cars parking in a car park such as the number of cars arriving in a given time interval, the duration a car stays in a car park, the relation between a car park and its surrounding business environment etc. To get these real data, we proposed to Media Research Consultant (MRC) who is awarded a Study project from LTA collecting car park data from 65 locations in Singapore, including 30 retail malls, 15 F&B clusters, 10 hawker centres and 10 community clubs, and we promised to help them compiled 5 site study reports as return for accessing the data they have collected.

3. Objectives

This project aims to complete two tasks:

1. To study Singapore public car parks parking situation
2. To build an agent-based model for Car Park designers and engineers to simulate and evaluate the efficiency of the car park by simulating a car park in different scenarios determined by a series of variables

Also, the project comprises of 2 phases:

1. Preliminary Phase focused on delivering the reports requested by our project sponsor – Media Research Consultants and find out the underlying issues in studied sites
2. With pre-processed data being ready to use, Phase 2 covers the development process of the agent-based model in NetLogo as well as the evaluation of the model's performance.

4. Phase 1 – Report Compilation

During the Preliminary Phase, we delivered 5 final reports of different development sites to our project sponsor. The reports served as a reference for LTA to understand the current parking situation of the sites. Necessary data and infographics have been collected and prepared by our sponsor such as parking occupancy surveys, human traffic counts and on-site interview surveys, while our team's tasks were interpreting the surveyed data, generating useful insights with regards to the parking situation and finally compiling the analysed results into the report with the use of data tables, charts and infographics.

A typical final report structure contains the following components:

1. Executive Summary
2. Site Background
3. Site Characteristics
4. Site Assessment
5. Survey Findings
6. Conclusion

Refer to Interim Report for the details of the content of each component.

By the end of Preliminary Phase, our team has finished 5 final reports as planned at the beginning of the project. Namely,

1. Greenwood Ave & Hillcrest Rd
2. Vivocity
3. Frankel Ave
4. Jalan Mata Ayer
5. Yuhua Market & Hawker Centre

5. Phase 2 – Model Development

5.1 Objectives

Being the primary phase of the project, Phase 2 aims to develop an agent-based simulation model using Netlogo to simulate the occupancy rate in a car park with different settings. The model will achieve a certain degree of flexibility, to take into account different inputs from the user and generate realistic outcomes of the usage of the car park for the user. This will allow further study on the validity of the design of the car park and subsequently, make necessary adjustments.

5.2 Literature Review

In the paper "*Agent-Based Model of Driver Parking Behaviour As a Tool for Urban Parking Policy Evaluation" (Itzhak Benenson, Karel Martens, Slava Birfir, 2007)*, a tool with similar function was developed. The tool aims to help policy makers explore the impacts on car drivers' parking choices by accounting of car parks' prices against the distance preference from destination of drivers. The paper showed us a good way of modelling drivers' behaviour in selecting parking places. The model quantifies the searching distance a driver is willing to travel against the driver's estimated fraction of free parking places.

Additionally, in the paper *"Car Parking as a Game Between Simple Heuristics" (John M. C. utchinson Carola Fanselow Peter M. Todd, 2012),* it explains the possible modelling algorithms for parking within a car park which are very helpful for us to understand the similarities, differences and necessary considerations during the selection of parking algorithms. Our group managed to use a combination of algorithms described in the paper to model ra eal driver's behaviour and reduce the error rate of the model. The details of the parking algorithm will be further described in Chapter *5.4.3 Parking Algorithm.*

Combining the approaches in these two papers, we have a good heuristic algorithm to simulate both the routes and the parking lot a driver tends to select in our modelling.

5.3 Project Methodology

Our group decided to simulate the issue using agent-based simulation by developing a simulation model on Netlogo. The reasons for choosing this approach will be elaborated in detail in the following sections.

Agent-based Simulation is a micro scale model that simulates the simultaneous operations and interactions of multiple agents in an attempt to re-create and predict the appearance of complex phenomena. Agents are a defined class of objects which have certain attributes and behaviours, they will interact with each other and also the world settings within the environment based on their pre-defined attributes and behaviours. Agent-based simulation is commonly used when agents' behaviour can be easily defined by simple behavioural rules but the outcome could be complex and difficult to predict, such as the modelling of reproduction, economic benefits, social status and others.

5.3.1 Why Agent-Based Simulation

There are 3 main benefits for agent-based simulation compared to normal simulation, such as business process simulation. It is the most natural way to model certain issues and helps visualize the phenomena in real time. In addition, it is flexible and can easily be adapted to new constraints.

The most important reason for choosing agent-based simulation is that the utilization of the car park depends on every individual drivers' behaviour, it is almost impossible to simulate the problem on a Macro level since we cannot coordinate and predict everyone's behaviour in the big environment.

Therefore, the approach from a Micro level is more applicable in this case since we can simply define individual's behaviour and allow them to interact freely to simulate the problem and generate the results. Hence, we have decided to apply agent-based simulation on the car park issue and in this case, the agents represents the drivers parking their cars. By defining individual agent's behaviour based on the data collected, we can simplify the issue. When we have a model which contains thousands of rational drivers within a certain car park size, their interactions will tell us whether the car park has enough parking lots and how big is the overspill.

Additionally, visualization which is commonly used in agent-based simulation helps us better understand which area of the car park tend to have congestions. This happens in real time along with the time axis. The user will be able to link the congestions with different periods of the day, not only logically, but also visually.

Lastly, the model will be flexible enough to takes in user's inputs since the drivers' behaviour can easily be adjusted for once and all the drivers will have the updated behaviour. Similarly, the environment can be adjusted to the user's preferred settings and a more realistic outcome can be generated.

5.3.2 Why Netlogo

The agent-based simulation community has developed multiple power tools over the past years which cater to different contexts. Our group chose Netlogo over the others due to a few reasons, user-friendliness, flexible in deployment environment, easy for maintenance and customization.

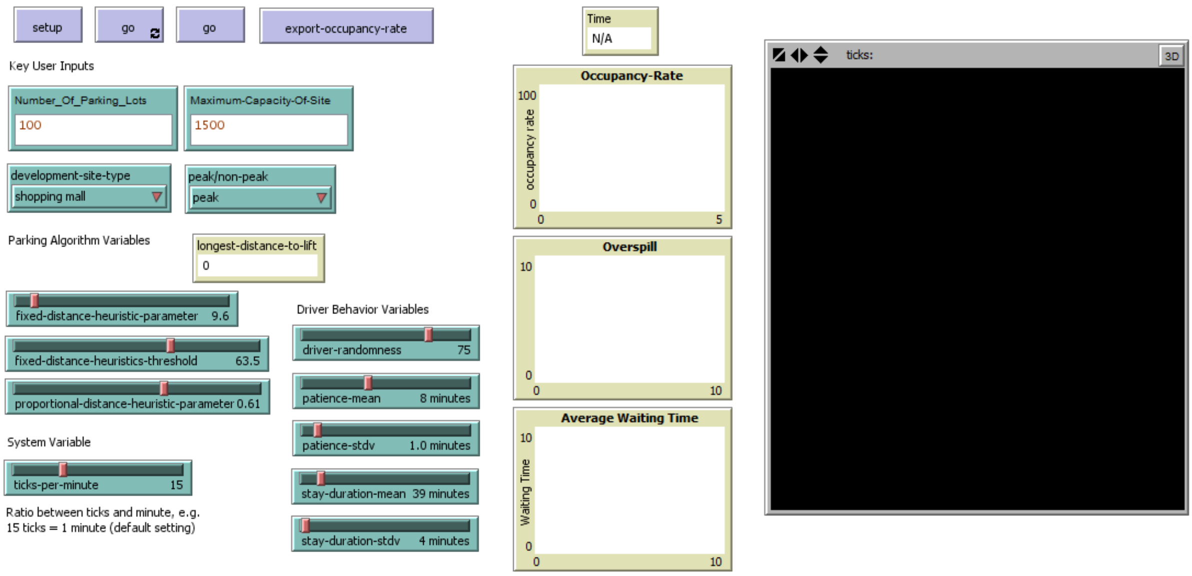


Figure 1: User Interface

Firstly, Netlogo is a great tool for users who are not familiar with IT tools and only wished to study certain social or economic issues. Netlogo was previously designed for statistician who would like to study social issues without understanding much programming knowledge, therefore, the tool was designed to have a very straightforward interface with buttons, slides, dropdowns and an environment called “world”. Any users who wish to learn to use a model will not take more than a few minutes to understand the functionalities of different items in the interface. Therefore, the tool will be convenient for our clients to use as they probably do not have much IT background.

Secondly, Netlogo only requires one time installation which can be done on various platforms and operating systems. The user will only be required to prepare the model file and open it in Netlogo to start the simulation. This is very convenient and does not require any prior learning from the user.

The last reason is equally important as Netlogo uses a simple coding language which can be learnt easily even for non-IT professionals. The coding language follows natural phrasing of our daily language to ease the learning process, so if there are any modifications or upgrades which needed to be done, the future developer can easily customize the model to suit their own objectives. This provides sustainability to the model and also extends its capabilities.

5.4 Data Pre-Process

The car park usage data can be naturally grouped into 4 categories based on sites’ nature: Shopping Mall, Community Centre, Hawker Centre and F&B Cluster. In pre-processing data phase, we examined our assumption that the patterns of cars arriving in a car park are homogeneous as long as the car parks belong to the same category site.

Knowing the pattern of cars arriving in a car park associating with the timeline and site category is critical as it is the input for the Agent-based model we built and will help define the behaviour of car agent in the model.

5.4.1 Data Cleaning

1. We did data cleaning in all 4 categories: Shopping Mall, Community Centre, Hawker Centre and F&B clusters and the process is as follows:
    I. Car inbound counts of each site during weekday and weekend are first collected from individual data file and mapped on a common timeline from 7am to 10pm.
    II. As each site data may be collected at different time period in a day, there are blank fields where no data being collected during the time intervals, and those fields are kept as it is to keep the data integrity.

III. Maximum human traffic during the peak day of the site is collected from individual data file as reference to the capacity of the site

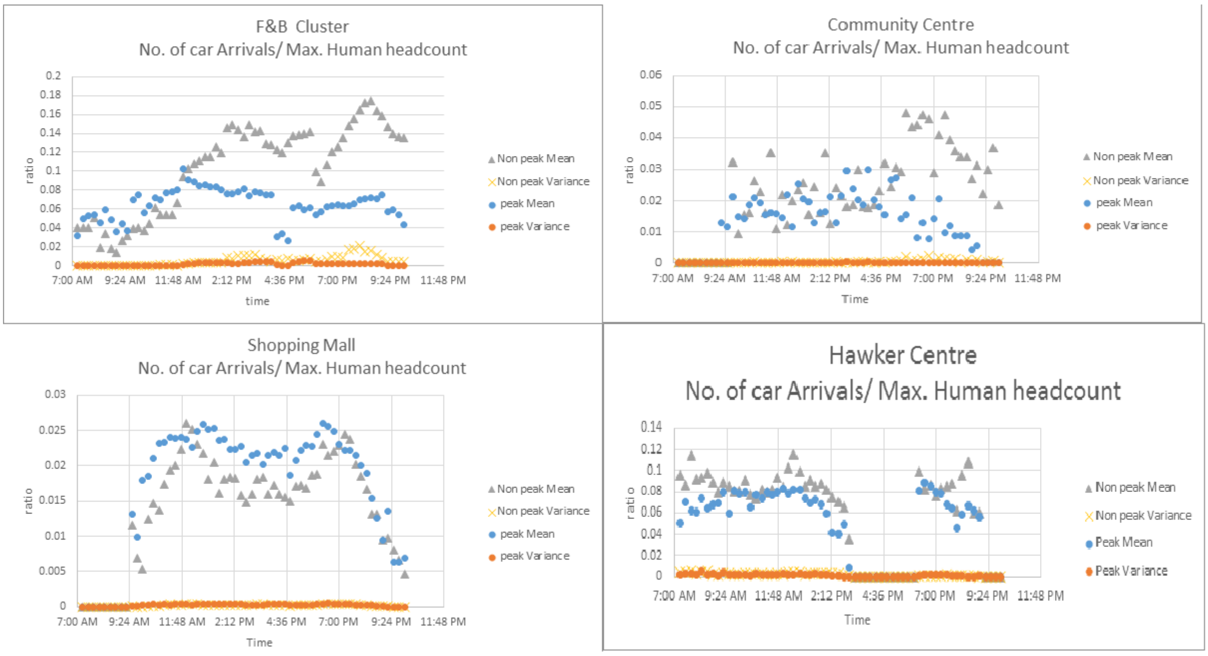
2. Base count at the start of the day is assumed to be 0
3. Net in and out human traffic is added to or deducted from the base count
4. Only the maximum value will be recorded as reference to the capacity of the site
5. Car counts of the site are then normalized by using car count at each timestamp divide by the maximum human traffic count, the result value can be defined as "number vehicles parked per human count at the time stamp"
6. Normalized car counts of all sites within the same group are then grouped together
7. Distribution of the car count at different time intervals is calculated to get mean and variance

The table below gives an overview of the compiled data:

	Shopping Mall	**Community Centre**	**Hawker Centre**	**F&B cluster**
Number of Samples	17	10	10	7
Time Duration	10am - 9pm (weekday & weekend)	9am - 6pm (weekend) 2pm - 10pm (weekday)	7am -3pm (weekday & weekend) 6pm - 8pm (weekday)	10am to 10pm (weekday & weekend)

5.4.2 Compiled Data Distribution

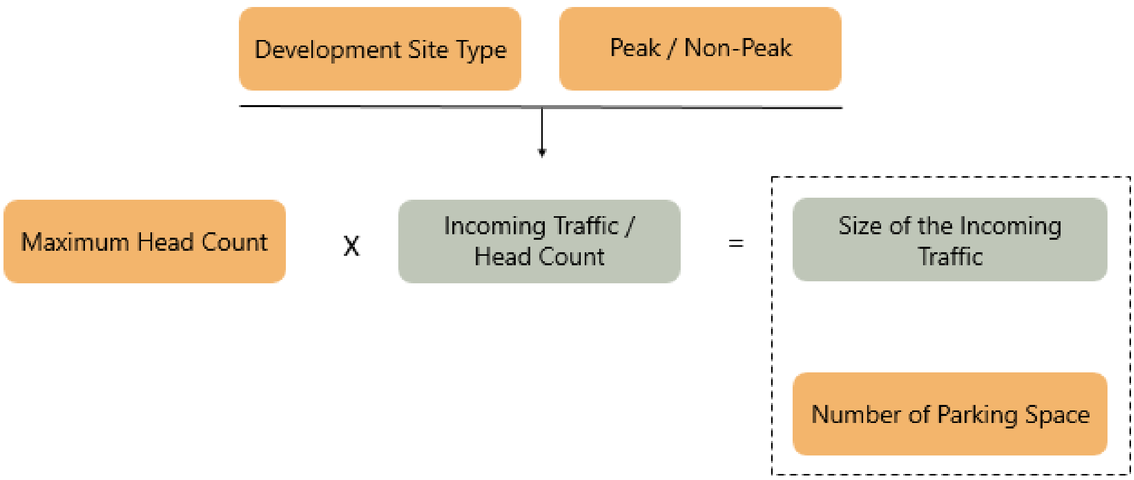
Data pattern of a few sites deviates greatly from the one of the rest sites in a same category as indicated by the variance at different timestamp. The deviation is mainly due to the popularity of the particular sites and it does not represent the trends of majority sites in the category. Therefore, the data of teose sites should be removed. After filtering out these data, the trend of car arriving on a weekday/ weekend in 4 different category sites are shown in following charts.



Alternative approach to maintain data integrity without filtering out "outliers" could be adding a parameter denoting the confidence interval for user to select. Based on user's selection, the model takes different sets of data into process. This approach will be discussed in Further Development section.

5.5 Agent-based Model

5.5.1 User Inputs and Parameters



There are 4 key user inputs which have to be specified before the simulation of the model. They are "Development Site Type", "Peak/Non-Peak", "Maximum Head Count" and "Number of Parking Space" as highlighted in orange in the figure above.

Based on “Development Site Type” and “Peak/Non-Peak”, we will retrieve the processed data that belongs to the 2 categories and this data will be used to calculate the “incoming Traffic / Head Count” at different time periods of the day. Next, the user will use “Maximum Head Count” to specify the rough size of the development site, for example, 1000 indicates that the development site will hold roughly 1000 persons at maximum. We multiply the head count with the incoming traffic to find out the “Size of the Incoming Traffic” which is relative to the size of the development site. Lastly, we will generate the cars at different periods based on this data, the cars will interact with each other and also the world to see whether the “Number of Parking Space” entered by the user is enough for such development site.

The following tables shows a list of parameters on our model’s interface and their respective functionality:

User Input/Parameter	Function	Remark
Number of Parking Lots	Configure the car park capacity	
Maximum Capacity of Site	Configure the maximum patron’s headcount in associated development site (e.g. shopping mall)	A car park inbound traffic in the simulation model is positively related to the headcount of associated development site.
Development Site Type	Select one of the four types of development site that the car park is associated with	
Peak/Non-peak Day	Choose to simulate Peak day traffic or Non-peak day traffic	
Longest Distance to Lift	Display the longest distance to the lift for reference so that the user can decide a suitable Fixed Distance Heuristic Parameter	Parking Algorithm Variable
Fixed Distance Heuristic Parameter	Set the distance range from lift which is considered as optimal parking position by drivers in a car park, any vacancies with the distance shorter than the parameter will be potential choice for the driver	Parking Algorithm Variable
Fixed Distance Heuristic Threshold	Set the threshold value for Fixed Distance Heuristic Algorithm, the driver will switch to Proportional distance Heuristics if the occupancy rate is higher than the threshold	Parking Algorithm Variable
Proportional Distance Heuristic Parameter	Set the proportion P value for Proportional Distance Heuristic Algorithm, user will use the P value to multiple the distance between the fixed park car he encounter and	Parking Algorithm Variable

	the destination. Any vacancies with the distance shorter than the result will be potential choice for the driver	
Driver Randomness	Randomness define the possibility a driver does not follow the heuristic rule defined	Driver Behaviour Variable
Patience Mean	Patience values are subjected to a normal distribution. This is to configure the mean value of the distribution.	Driver Behaviour Variable. Patience is a time threshold value beyond which a car will stop searching for its expected optimal parking position, instead, it will park in any available lot near it.
Patience Stdv.	Patience values are subjected to a normal distribution. This is to configure the standard deviation value of the distribution.	Driver Behaviour Variable
Stay Duration Mean	Configure the mean value of the distribution of time duration a car would park in a car park.	Driver Behaviour Variable. The distribution is assumed to comply with a normal distribution.
Stay Duration Stdv.	Configure the standard deviation value of the distribution of time duration a car would park in a car park.	Driver Behaviour Variable. The distribution is assumed to comply with a normal distribution.
Tick Per Minute	Configure the conversion unit between ticks and minutes	System Variable. Tick is the time unit for an agent to process one action.

5.5.2 Agents (Patches and Turtles)

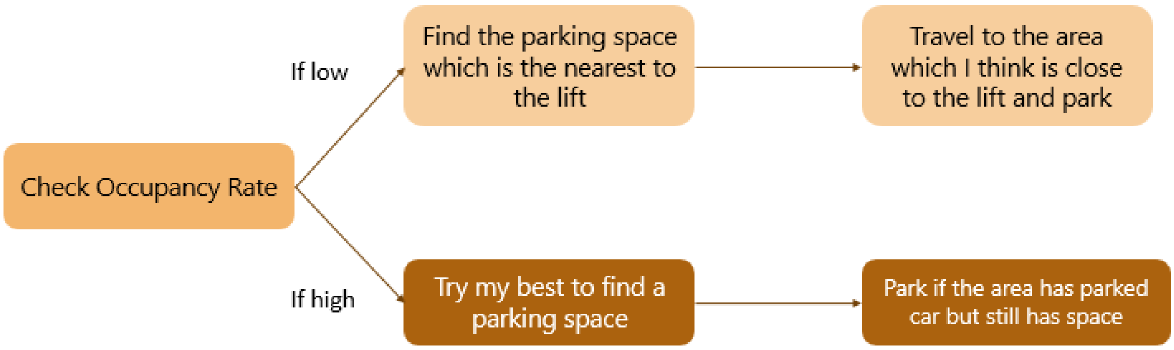
There are 2 types of agents in our model, patches and turtles. Patches refer to the group units in the environment which are used to facilitate turtles' behaviour, and in this case, we have 3 patches, "Road", "Parking Space" and "Wall". On the other hand, turtles refer to a class of agents which travels and interacts in the environment, and 1 turtles class, "Car" is implemented in our model.

The attributes for patches are relatively easy to describe. "Road" is used to allow cars to move around, but additionally, some of the "Road" are identified as "Entrance" which only allows "Cars" to enter into the carpark, "Exit" which only allows "Car" to exit from the carpark and "Crossroad" which allows the cars to make turns. "Parking Space" is used to identify parking lots in the carpark, it will be labelled as "yes" to park if the lot is not occupied by a "Car", the status will change to "parked" if the lot is occupied by a "Car". "Wall" is simply used to define the boundary of the carpark and the cars will not be able to drive pass "Wall" patches.

“Car” is slightly more complex to define. It has 4 basic behaviours, firstly, it must follow basic traffic rules, in our context, a car must drive on the left of the road and only make turns at crossroad. Secondly, a car or a driver will try to find the optimal lot to park the car until driver’s patience runs out, we use “Patience” variable to describe the car or driver’s patience, it is modelled using normal distribution with user’s input of its mean and standard deviation. If the patience of the driver runs out, he will choose to park whenever he encounters the first free parking space. Thirdly, the driver will leave the carpark after the “Stay Duration” is reached. Similarly, “Stay Duration” is modelled using normal distribution with user’s inputs. Lastly, we introduce the idea “Randomness”. Since not all the drivers will follow the optimal parking algorithm, different drivers may make different decisions based on their own preferences. Hence, we give the driver a “Randomness” variable to create a more realistic decision making process.

5.5.3 Parking Algorithm

Before discussing the detailed parking algorithm, we have to first understand the drivers’ behaviour during normal daily life and transform it into an algorithm which best describes it. This is to achieve more realistic and accurate simulation result.



The figure above describes a driver’s decision-making process when entering a car park. He will have a rough idea of the occupancy rate in the car park since most of Singapore’s car park shows the number of vacancies at the entrance. If he thinks that occupancy rate is generally low, he will prefer to find a parking space which is nearer to the destination, in this case, the lift or the shop that he wants to go, so as to minimize the walking distance. The immediate action will be to travel to the area, which he thinks is close enough, and check if there are any vacancies and henceforth choose any one of them. On the other hand, if he thinks the occupancy rate is high, the strategy is to find a parking space, which is as optimal as possible. In this case, if a vacancy can be found in an area with many parked cars, it is thought to be a good choice since many people choose to park here while the occupancy rate is high.

With these 2 simple behaviors in mind, we want to find parking algorithm which suits them the best. After reading through several research papers, especially in the paper *"Car Parking as a Game Between Simple Heuristics" (John M. C. utchinson Carola Fanselow Peter M. Todd, 2012).* We managed to find a few possible parking algorithms to implement:

	Fixed-Distance-Heuristics	**Proportional-Distance-Heuristics**	**Car-Count-Heuristics**	**Space-Count-Heuristics**	**Block-Count Heuristic**	**X-out-of-y-Heuristics**
	Takes the first vacancy encountered within a fixed distance	The first vacancy after driving a proportion P of the distance between the first occupied place encountered and the destination	The first vacancy after passing C parked cars	The first space after reaching the first parked car and then passing S available spaces	The first space after passing a block of at least B parked cars without a space	If x or more parking places were occupied out of the last y places passed
Require Knowledge of Destination?	Yes	Yes	No	No	No	No
Suitable for Our Model?	Yes (if occupancy is low) May be affect by Carpark Size	Yes (if occupancy is high)	Yes (if occupancy is high) May be affect by Carpark Size	Yes (if occupancy is high) May be affect by Carpark Size	Yes (if occupancy is high) May be affect by Carpark Size	Yes (if occupancy is high) May be affect by Carpark Size

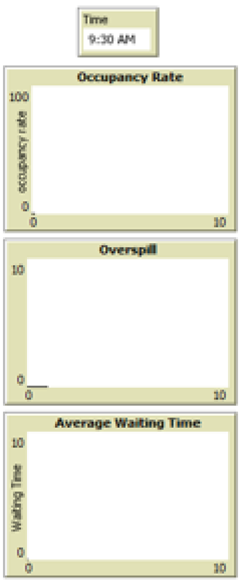
The brief explanation of each algorithm is stated in the second row of the table. We will explain "Fixed-Distance-Heuristics" and "Proportional-Distance-Heuristics" in detail since they are the selected algorithms in this study. "Fixed-Distance-Heuristics" means that the driver will take the first vacancy encountered within a fixed distance to the destination, it takes in a parameter "Distance" which represents the maximum distance that the driver can accept in order to choose the vacancy. "Proportional-Distance-Heuristics" means that the driver will record down the distance between the first parked car he encounter to the destination, a parameter "Proportion P" will be used to multiply that distance and any vacancies which are smaller than the result will be chosen by the driver. There are 2 main reasons for choosing the 2 algorithms; firstly, both algorithms take into consideration the distance to the destination, which we believe is very important in the car park study. The rest of the 4 algorithms do not consider this piece of information and hence might not describe the real case scenario accurately. Secondly, in terms of matching the driver's behavior, "Fixed-Distance-Heuristics" is very close in the case when the occupancy rate is low while all the rest of the algorithms can be used when the occupancy rate is high. However, one main difference for "Proportional-Distance-Heuristics" is that its parameter is relative to the car park size while the other algorithms request a different input when the user change the car park size. Therefore, we can use the optimal value for the "Proportion P" in the simulation with varying car park size. This will definitely increase user-friendliness and accuracy of the model. With these considerations, we have chosen "Fixed-Distance-Heuristics" and "Proportional-Distance-Heuristics" to be the parking algorithm.

However, drivers in real life do not strictly follow the optimal solution all the time, which is why we introduce the idea of "Randomness" and "Patience" which have been described earlier in the user inputs and parameters sections. The model will check the driver's patience before deciding the parking algorithm, if the patience runs out, the driver will choose the nearest vacancy. If not, he will proceed to choose either "Fixed-Distance-Heuristics" or "Proportional-Distance-Heuristics" based on the occupancy rate of the car park. After choosing the algorithm and decide on the direction to go, there is a chance which the driver may turn randomly instead of following the result direction from the algorithm since we want driver's behavior to be more random and more realistic. This level of randomness can be adjusted by the user. This loop will continue until the driver finds a parking space and start counting his "Stay Duration".

"Stay Duration" is used to model that amount of time which the driver will stay in the car park, we used normal distribution to randomly generate a "Stay Duration" for the driver based on user's input of mean and standard deviation. The driver will leave the car park when the "Stay Duration" is reached and this will end the whole parking process for the driver.

5.5.4 Results Monitoring

The performance of the car park is monitored using 3 indicators: occupancy rate, overspill and average waiting time. Occupancy rate is calculated by using the number of parked cars divide by the number of parking spaces. Overspill counts the number of cars which are either waiting at the entrance or trying to find a parking space when the occupancy rate is 100%. Average waiting time takes the average of all cars' waiting time which are not parked.



These 3 indicators are collected in real time and plotted on a graph. The data can be exported into csv file and further study can be done if the user would like to study the data in detail.

5.5.5 Model Evaluation

The accuracy of the model is hugely important, we want the model to provide realistic simulation of the car park usage so that the user can trust the results and make necessary adjustments based on the simulation results. Therefore, we randomly picked 2 development sites from each category and use their parameters to run the simulation, the simulation results were then exported to compare with the real data. We took the absolute difference between the simulation results and the real data and divide it by the real data to get a percentage error.

The table below shows the results:

	Shopping Mall	Hawker Center	Community Centre	F&B Cluster
Peak	23.50%	37.64%	27.59%	35.30%
Non-Peak	29.82%	31.08%	31.36%	38.29%

Generally, Shopping Mall and Community Centre have better accuracy rate compared to the other 2 development sites. The reason is probably that for Hawker Centre, the human count is quite unstable which can affect the incoming traffic size and eventually the occupancy rate. For F&B cluster, the car park tends to be very small which usually has less than 50 parking space, few cars difference can lead to large simulation error in terms of percentage.

However, our group feels that the error rate is acceptable because the simulation for each development site type took the mean value for incoming traffic, there will definitely be some error when comparing the mean with an individual data point. Therefore, the user can trust the simulation results and make informed decision in terms of car park usage.

5.6 Project Assumptions and Limitations

Since our project was done based on a few assumptions, the model also has its limitations, this section will highlight the major points:

- **No real data of parking duration**

  One of the major limitations is that there is no support of real data of parking duration. In our case, we use a normal distribution with user specified mean and standard deviation to generate the “Stay Duration” for each driver. However, this will definitely affect the accuracy of the model.

- **Constant speed of cars**

  The speed of the car was designed to be constant which is not true in real life, this could affect the amount of time which the driver takes to find a parking space.

- **Lack of details of driver behaviour (e.g. enter, exit and parking action is instantaneous)**

  Similar to the previous assumptions, there could be many more detailed driver actions, such as entering and exiting the car park, the parking process and others. All of these will affect the traffic in the car park and hence car park usage.

- **Choice of parking algorithm**

  The parking algorithm for our model was chosen and modified based on common understanding of driver's behavior in the car park and his decision making process. However, the real case could be different or more details need to be considered.

- **Pre-designed Car park Shape**

  In order to maintain certain flexibility, our model chose to build the car park into a pre-designed car park shape so that we can adjust to user's input of car park size. However it is also understood that car park shape will affect the cars' behaviour significantly.

5.7 Further Development

The simulation program can be further developed with the introduction of the following features:

- **More detailed and accurate behaviour of drivers**

More detailed parking cars' parking behaviours can be studied and modelled, such as the different parking styles: Angle Parking, Perpendicular Parking, and Parallel Parking etc.

In our model, due to lack of data describing the parking duration of cars, we set its configuration as adjustable parameters that allow user to manipulate. Improvements can be made by collecting and modelling with realistic data showing the pattern of cars' parking duration.

- **Map realistic car park layout into the model**

Modelling with realistic car park layout allows user to get simulation results that are more representative of the real traffic outcomes in the car park. It can be achieved by importing the raster map into NetLogo. Pixels in raster data have to be assigned with appropriate attribute values so that it allows programmer to define the rules for each patch in NetLogo.

- **Introduce additional parameters**

We have identified two kinds of parameters that will improve the accuracy of our model:

1. Configure different range of confidence interval in defining car arriving pattern
2. Define the attractiveness of car park associated development site. Besides, the model should capture the relation between this attractiveness parameter with the number of inbound traffic in the car park.

6. Conclusion

The first phase of this project focuses on providing data analysis and reports compilation for the sponsor. The second phase of the project focuses on building an agent-based simulation model which can be reused with flexible settings to study car park usage and overspill, which is also the main focus of the overall project.

The team managed to develop an agent-based simulation model on Netlogo which can take in parameters from users to simulate real life car park usage, the results were proven to be quite accurate and can be used by the users to make adjustment in terms of car park design. This tool will be very helpful in evaluating car park performance or even future car park planning. Moreover, the tool is very user-friendly so that our target clients who are probably car park designers from different agencies with minimal IT background can use the tool with little learning. However, there are definitely some limitations to the simulator, further development will focus on collecting data of driver stay duration and building a more detailed driver behaviour in order to achieve higher accuracy.

7. References

1. Agent-Based Model of Driver Parking Behavior As a Tool for Urban Parking Policy Evaluation, (Itzhak Benenson, Karel Martens, Slava Birfir, 2007)
   https://agile-online.org/Conference_Paper/CDs/agile_2007/PROC/PDF/91_PDF.pdf

2. PARKAGENT: An agent-based model of parking in the city (Itzhak Benenson, Karel Martens , Slava Birfir, 2008)
   http://www.tau.ac.il/~bennya/publications/CEUS2008BenensonMartensBirfir.pdf

3. Comparison of agent-based modeling software
   https://en.wikipedia.org/wiki/Comparison_of_agent-based_modeling_software

4. AGENT-BASED MODELS methodology and philosophy
   http://www.agent-based-models.com/blog/resources/simulators/

5. NetLogo: Where We Are, Where We're Going (Paulo Blikstein, Dor Abrahamson, and Uri Wilensky)
   https://ccl.northwestern.edu/papers/BlikAbrWil_IDC-05-demo.doc.pdf

6. Netlogo simulation: effective for agent based modeling in artificial intelligence, Gargi Kanjilal, Bhagyashree Rajgire, Veerendra Kumar Jain (M.Tech Software Technology)
   http://www.ijritcc.org/download/IJRITCC_1333.pdf

7. Car Parking as a Game Between Simple Heuristics , Hutchinson, J. M. C., Fanselow, C. & Todd, P. M. (2012)
   http://www.indiana.edu/~abcwest/pmwiki/pdf/hutchinson.car%20parking.2012.pdf