Project 1 Final Report

Hiep Le – CS 150 – 03/20/2018

1. Introduction

The project aims to design a simulation so that we can determine the optimal combination of features for an amusement park. The park will have rides of different types and each ride will have its own attractiveness, length of time and other features. In our simulation, we make the following assumptions (Liew, 2018):

- All visitors pay a single price for each individual
- It takes one timestep to load a ride
- The same distribution of visitors will arrive throughout the day
- Each person takes up one space
- The cost for the park is the total space taken up by the rides (and the queues)

There are also several rules governing the simulation (Liew, 2018):

- There is no space required for moving between rides as movement between queues is instantaneous.
- Each person will only try each ride either 0 or 1 times, groups will ride together.
- Older people can go on rides for younger people but not vice versa. However, they cannot go on any rides designed for people that are younger than members of the group.
- The age groups that we have (in descending order): adults, young adults (18-22), teenagers (13-18), adolescents (5-12), toddlers (<5)
- Groups leave the park when either they have exhausted the rides they want, or there is no space for them to join the queue for a ride

2. Approach

Classes

The program consists of 14 main classes.

- RandomGaussian (Generate random numbers, n.d): generates random numbers from a normal distribution given a mean and a standard deviation. This class makes use of the class Random (Oracle, 2017) of Java API.

- Person: holds the age information of a person, which corresponds to a certain person type.

- Group: holds list of person objects. It holds information about the rides group can take and rides groups have taken. It also has methods to find rides for the group.

- Individual: subclass of Group class. Group only has one member in its member list. It is constructed with a randomGaussian for random age of its sole member.

- Family: subclass of Group class. Has at least two adults and a number of children of certain age. Both age and number of children are randomly generated.

- Friends: subclass of Group class. Has a number of person objects of certain age. Both age and number of members are randomly generated.

- GroupList: it is a group container, which allows us to call methods on all group members of the list.

- RideTemplate: holds information about a type of ride. These information is imported from input files and are used to construct ride objects.

- Ride: forms the basis of our park simulation. It has methods such as load, run, unload to simulate the working of the park. It also changes the variables and information of groups which have joined the ride using several void() methods.

- RideList: it is a ride container, which allows us to call methods on all ride objects of the list.

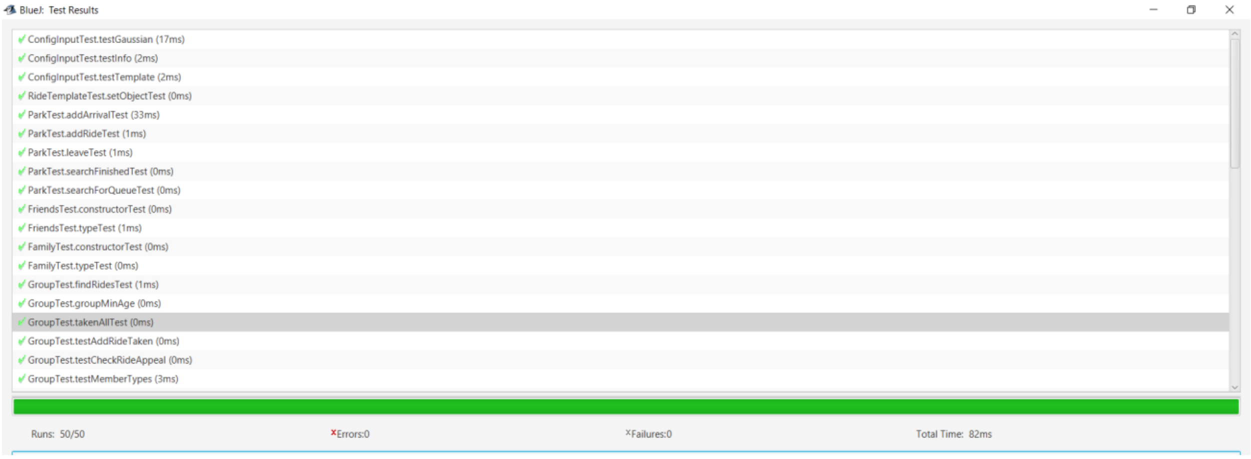
- Park: holds objects of RideList and GroupList. It handles the movement of groups between queues of rides in the list of rides and also the leaving of groups. It is also used to update time step for every groups and rides in the park.

- ConfigInput: it is used to read input from the config file.

- ParameterInput: it is used to read input from parameter file.

- Controller: handles the random arrival of new groups, taking in input files and running the simulation.

In addition to the aforementioned classes, we only have 12 test classes for every class aside from the Random Gaussian and Controller class. There is no test class for Random Gaussian because it is difficult to test the generation of random numbers. Figure 1 shows the results of the unit tests.



```
✓ ConfigInputTest.testGaussian (17ms)
✓ ConfigInputTest.testInfo (2ms)
✓ ConfigInputTest.testTemplate (2ms)
✓ RideTemplateTest.setObjectTest (0ms)
✓ ParkTest.addArrivalTest (33ms)
✓ ParkTest.addRideTest (1ms)
✓ ParkTest.leaveTest (1ms)
✓ ParkTest.searchFinishedTest (0ms)
✓ ParkTest.searchForQueueTest (0ms)
✓ FriendsTest.constructorTest (0ms)
✓ FriendsTest.typeTest (1ms)
✓ FamilyTest.constructorTest (0ms)
✓ FamilyTest.typeTest (0ms)
✓ GroupTest.findRidesTest (1ms)
✓ GroupTest.groupMinAge (0ms)
✓ GroupTest.takenAllTest (0ms)
✓ GroupTest.testAddRideTaken (0ms)
✓ GroupTest.testCheckRideAppeal (0ms)
✓ GroupTest.testMemberTypes (3ms)

Runs:  50/50          ×Errors:0          ×Failures:0          Total Time:  82ms
```

Figure 1. Unit Test Results

Design

In the simulation, we first set up a park of a number of rides of different types as specified by the input files. The rides will have different features that affect the behavior of the simulation.

When the simulation starts, groups will start arriving. The number and properties of the groups are randomly generated using parameters in the input files. The groups will then be added to the groups containers of the Park. They will then search for suitable rides in the list of rides of the park. The rides will then load, run and unload based on their time features and groups will move between rides when they have finished a ride. If they have taken all rides or there is no space, they will leave and be deleted from the group list of the park.

Algorithm and Data Structures

- ArrayList (Oracle, 2017) is used to store the groups and rides in the rides and groups containers since it is easy to manage. Insertion at the end is usually needed for both of our containers, which is fast for the List structure. Removal of groups usually requires searching through list to compare matching elements, which is done easily using ArrayList remove() method.

- HashMap (Oracle, 2017) is used in parameterInput class to store the number of rides for a ride template. The nature of the data, one value corresponding to another, makes the use of HashMap suitable.

3. Methods

The size of the park is fixed to be 600. The features of the different types of rides are also fixed (details in table 1).

The numbers of rides are then varied accordingly, with their combined size not going over the park size limit.

The simulation is set to run for 1000 timesteps. For each combination, the number of groups who visited, left and left early are recorded at the end. The number of visitors to each ride, the average and max waiting time are also presented at the end for comparison.

We will analyze the performances of each configuration based on several criteria:

- Number of guests who left early:
    - Number of visitors are constant for every case so the number of guests which left early are considered.
    - Groups which left early are defined as groups which have not taken all the rides they could.
- Profit: Because the number of visitors to the park are constant for every case, regardless of whether they take a ride, we will instead assume that each rider will pay a price to join a ride's queue.
    - Total revenue will thus be (the average riders per ride) x (number of rides).
    - The cost of the park is assumed to be the space taken by all the rides multiplied by 10. The 10 multiplication factor is necessary for more realistic cost/ revenue comparison.
    - Profit is thus the difference of total revenue to total cost.
- Average Waiting Time per Ride
- Max Waiting Time per Ride

Type	Roller Coaster	Ferris Wheel	Bumper Cars	Carousel
Groups	adult young_adult teenager	adult young_adult teenager adolescent toddler	adult young_adult teenager adolescent	adolescent toddler
Space	200	150	120	100
Length	3	10	10	5
Capacity	20	45	25	35
Queue	120	65	40	50

Table 1. Ride Type Information

4. Data and Analysis

Number of Guests Left Early Comparison

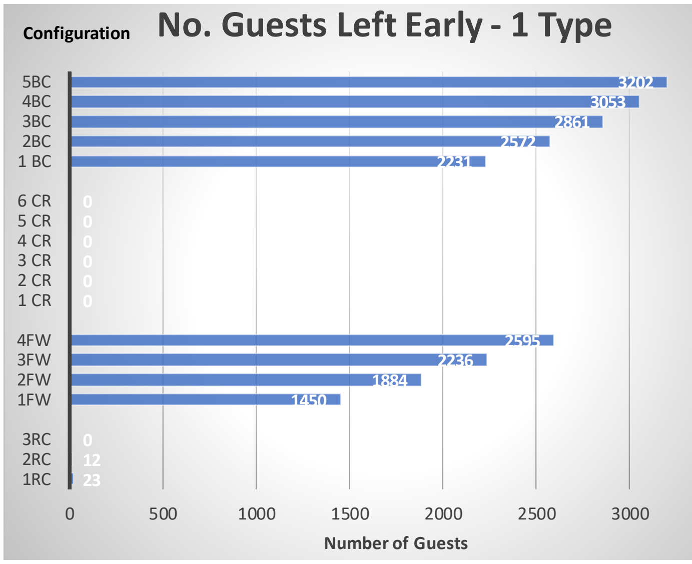


Figure 2: Number of Guests Left Early for One Type Park

For one type configuration, figure 2 shows that park of only Roller Coasters and Carousels have almost no early leavers. This is expected because their appeal types do not include all the types of person so not all groups will be able to join the line's ride. This reduces the number of riders on the queue, allowing the ride sufficient time to clear the queue and have space for new groups coming in the park. At the same time, groups which have no suitable rides to take will leave the park as soon as they enter. They will not be considered early leavers as they have the same number of rides can take and rides taken (zero).

On the other hand, parks of only Bumper Cars have the highest number of early leavers. This is because they attract almost all groups, have the longest ride length and shortest queue capacity. All visitors to the park will want to join the ride's queue. However, the ride does not end quickly enough to load new riders, leading to the queue capacity being maxed out. Visitors who arrive later will have no space to join the queues.

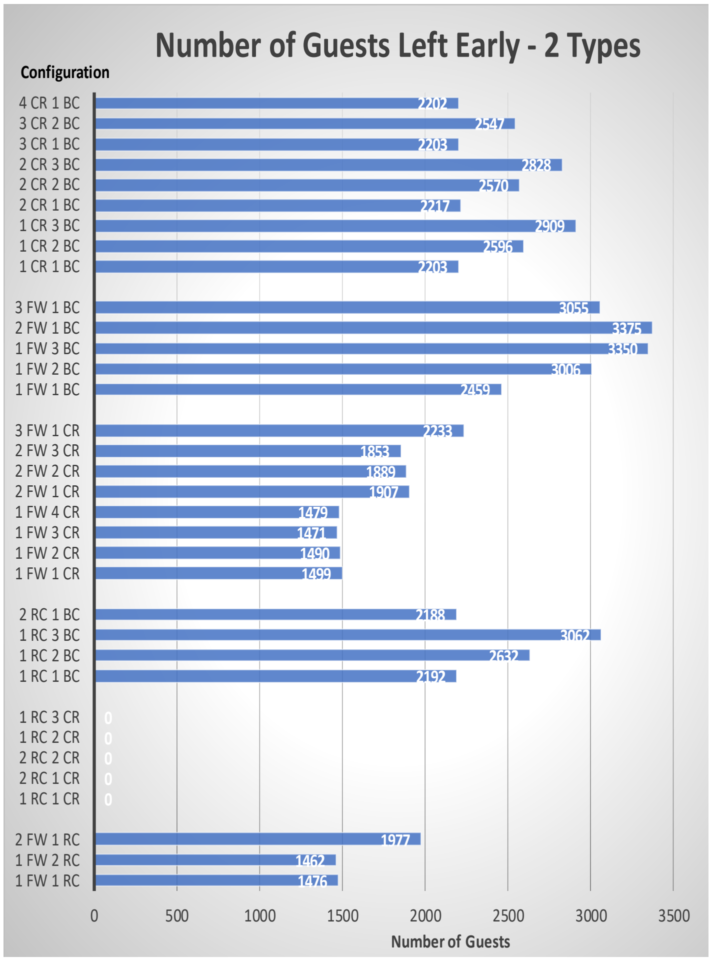


Figure 3: Number of Guests Left Early for Two Type Park

Data in figure 3 also supports the observation. The two – type configuration of Roller Coasters and Carousels have almost no groups leaving early. This is because they attract different groups and thus complement each other very well. Other combinations that work well are Ferris Wheel paired with Roller Coasters or Carousels. These types often have varying ride length, which allows the quicker ride to clear the queue while the longer ride is still running. This prevents the quicker ride from having a sudden surge of guests joining its queue.

The worst performance for figure 3 remains the combination of Ferris Wheel and Bumper Cars. These two types both have the longest ride length, attract all groups and have short queue length. The queue capacity will thus be filled quickly.

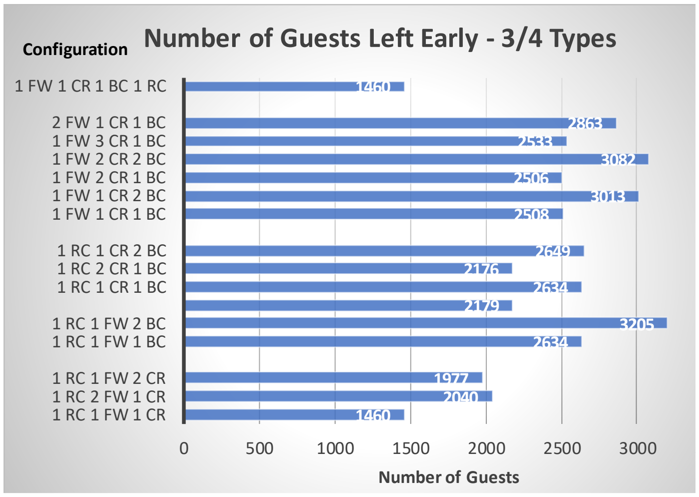


Figure 4: Number of Guests Left Early for Three/ Four Type Park

In figure 4, the configuration that leaves the lowest number of early leavers are 1 Roller Coaster, 1 Carousel and 1 Ferris Wheel. This combination provides a good mix of attracted groups and line size. Roller Coaster and Carousel attract different groups while Ferris Wheel can take all the riders that have finished either ride.

Additionally, Roller Coaster has a very short ride length along with the longest queue length, which means it can hold more riders in the queue, preventing them from leaving early. Ferris Wheel has the longest ride length, which also allows the other two rides sufficient time to clear the queue and have sufficient queue space for riders that just finish Ferris Wheel.

Figure 4 also suggests that the combination of four types gives a comparable number of early leavers to the best combination of three ride types. This can be explained with the fact that they have the highest number of queue size and varying ride length as well as attracted groups.

In general, to minimize number of early leavers, a configuration of rides with non-overlapping attracted groups is optimal.

Profit Comparison

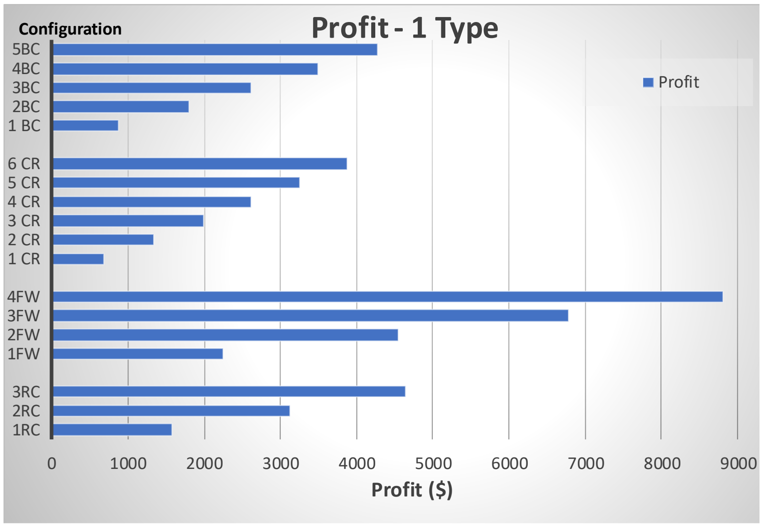


Figure 5. Profit for One Type Park

In one type park configurations, figure 5 shows that configuration of 4 Ferris Wheels makes the most profit. This is because Ferris Wheels attract all groups and will thus have the highest of number of riders. Additionally, it has the largest ride capacity and thus can fit many riders into one ride, allowing for more space on the queue. On the other hand, configuration of only Carousels makes the least amount of profit as it attracts the fewest number of groups and thus not many guests would want to take the ride.

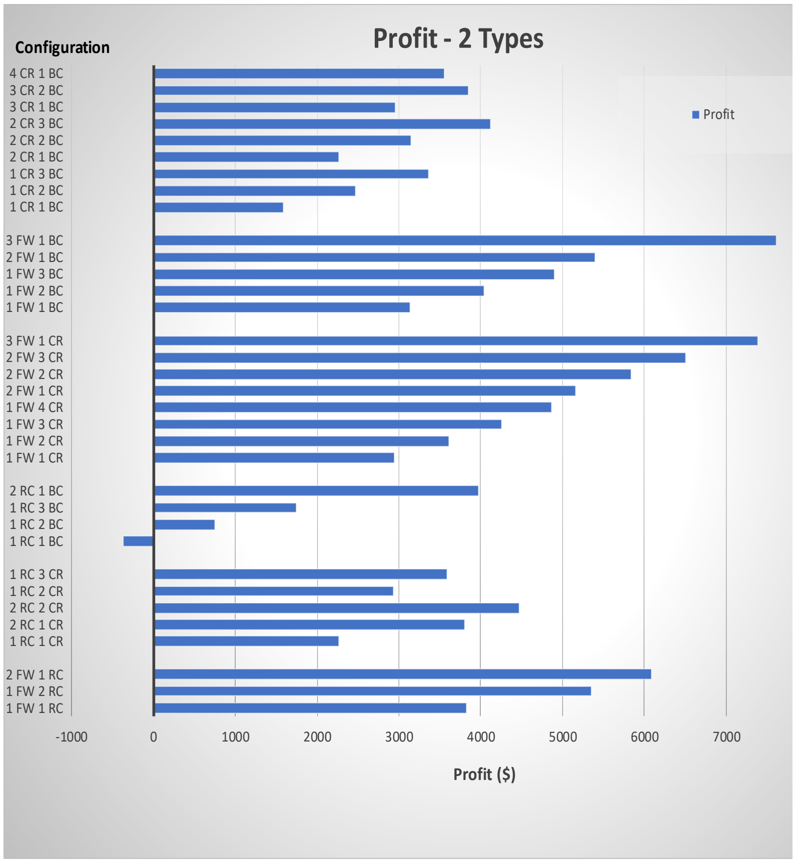


Figure 6. Profit for Two Type Park

Figure 6 also shows that the inclusion of Ferris Wheel in any configuration will greatly increase the amount of profits. In terms of maxing the profit, the configuration of 3 Ferris Wheels and 1 Bumper Carss gives the best performance. This is possibly because they both attract the majority of groups and will thus have very high demand.

However, in terms of maximizing average number of rides per ride, configuration of Ferris Wheel and Roller Coaster has the best performace. In figure 6, among all configurations of only 2 rides of 2 different types, the configuration of 1 Ferris Wheel and 1 Roller Coaster makes the most amount of profit. This is possible because Roller Coasters have very short ride length, which allows them to have the highest number of riders during the simulation.

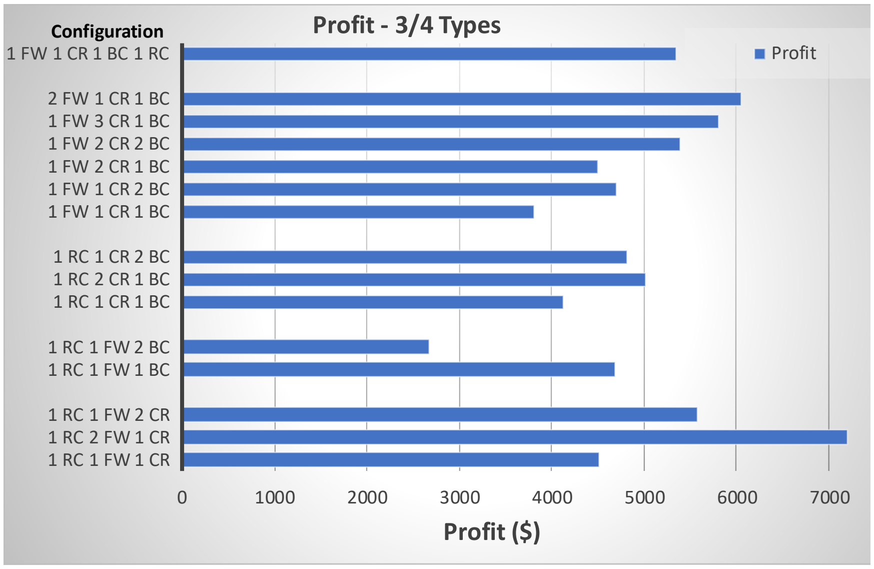


Figure 7. Profit for Three/ Four Type Park

It is thus reasonable that the 3-type configuration of Ferris Wheel, Roller Coaster and Carousel has the highest profit margin among all possible configurations (figure 7). We have already discussed Roller Coaster's ride length, but another factor could be the non-overlapping appeal groups of Roller Coaster and Carousel. They will help to reduce the number of visitors Ferris Wheel has to take at every time step. At the same time, the majority of guests are not attracted to the Carousel and as such the addition of Carousel do not increase the number of rides the majority of visitors have to take.

In general, profit will depend on the appeal groups of each type and whether they overlap or not.

Average Wait Time Comparison

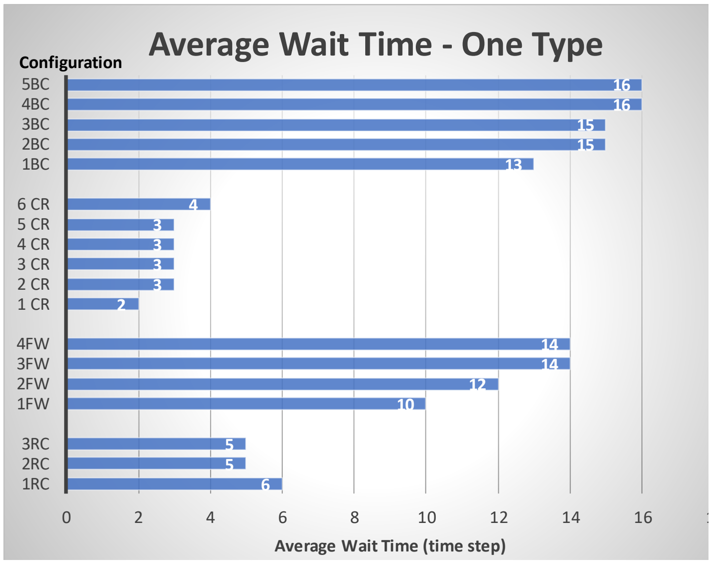


Figure 8. Avg. Wait Time for One Type Park

From Figure 8, we notice that the average wait time for configurations of Bumper Cars and Ferris Wheels are much longer than those of Roller Coaster and Carousel. This is because Roller Coasters and Carousels do not attract all age groups and thus have fewer groups joining their lines, leading to shorter average wait time.

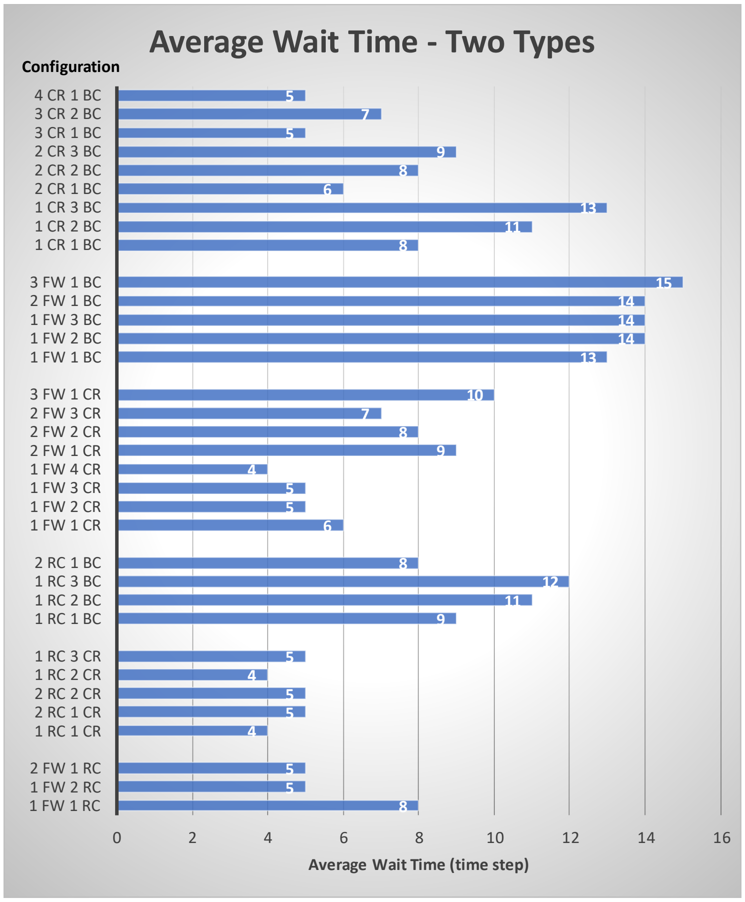


Figure 9. Avg. Wait Time for Two Type Park

The same observation applies to configuration of multiple types. Figure 9 shows that the configuration with the longest average wait time is that of Ferris Wheel and Bumper Cars. These two types both have long ride lengths and attract all age groups. This contributes to much longer queues and thus longer wait time.

On the other hand, the combination with the shortest average wait time is that of Roller Coaster and Carousel. This is understandable as they attract opposite age groups and thus their queues do not overlap, contributing to shorter queue length. Other combinations consisting of Roller Coaster tend to perform well on this metric as Roller Coaster has very short ride length and thus will load new riders from queue quickly.

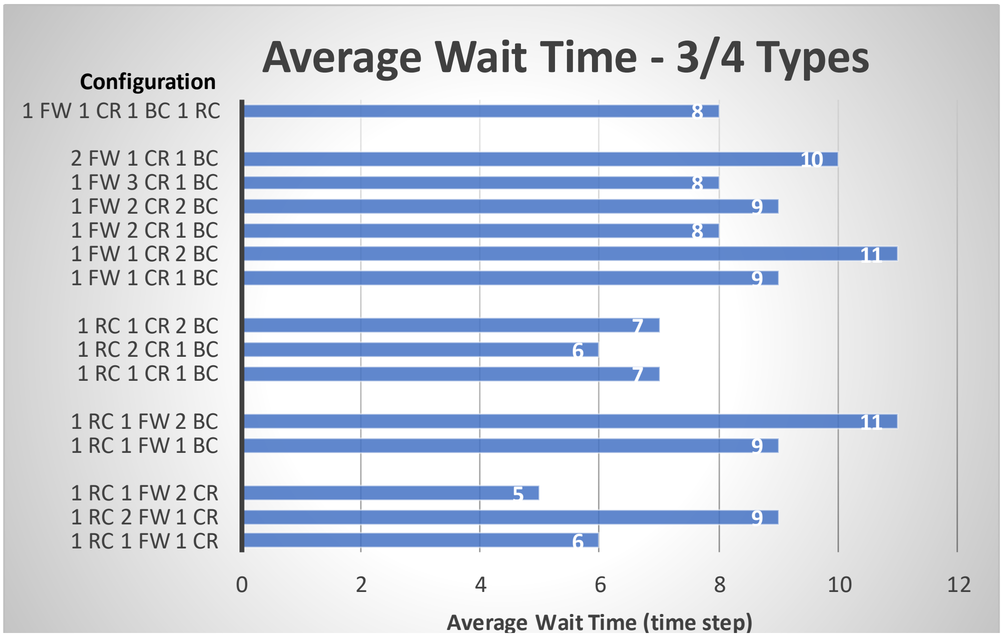


Figure 10. Avg. Wait Time for Three/ Four Type Park

From figure 10, the average wait time of any combinations comprising of Bumper Cars and Ferris Wheel will be long, also because of the longer ride length and the fact that they attract all groups. 4-ride configuration does not perform significantly better than 3-ride configuration. Because of the attracted groups overlap between the 4 types, visitors will want to take more rides and will stay in the park for a longer time, leading to longer average wait time.

In general, the average wait time for every combination depends on the length of the ride and the attracted groups for every ride. If we prioritize average wait time, the best configuration will be of rides with non-overlapping attracted groups, for example Roller Coaster and Carousel.

Max Wait Time Comparison

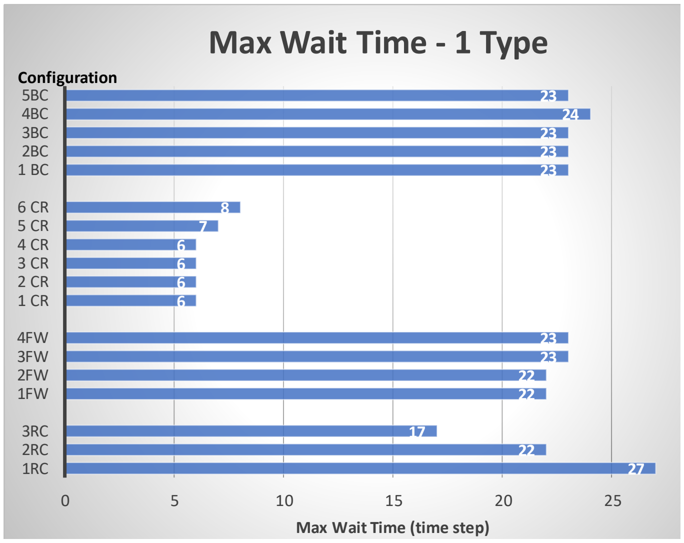
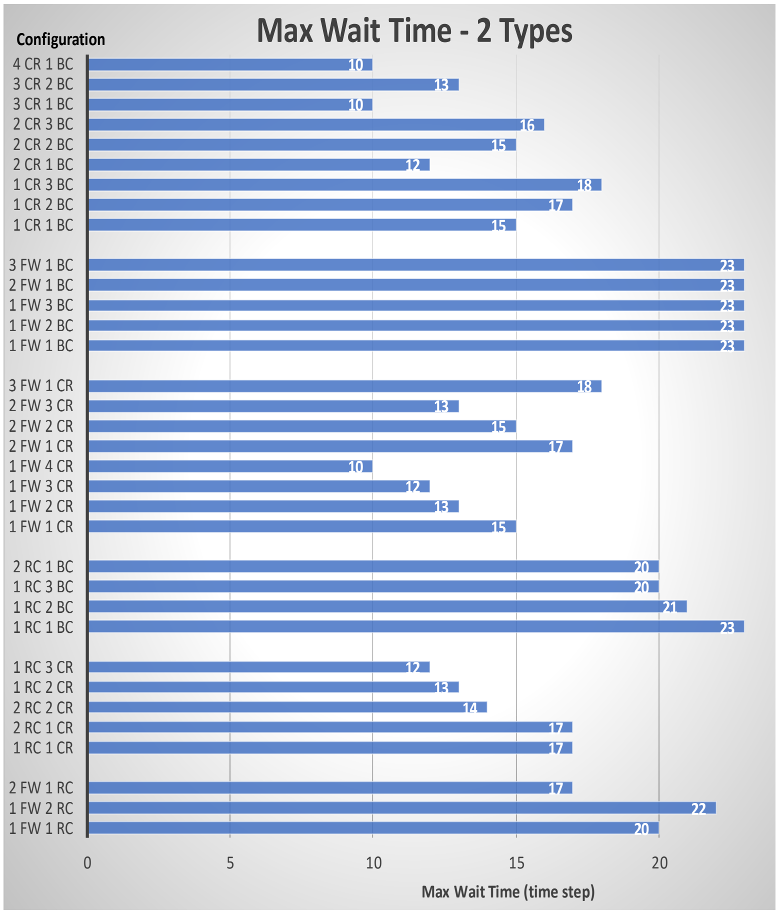


Figure 11. Max Wait Time for One Type Park

Figure 11 shows that configuration of only Carousels has the shortest max wait time due to the fact that it only attracts a small group of guests to the park. Thus, the queue will always remain relatively short compared to other configurations, resulting in the shortest max wait time. On the other hand, configuration of only Roller Coasters has the longest max wait time as it has the longest queue length, allowing more guests on the ride queue but also requiring them to wait for much longer time.



The same trend is observed in figure 12. Any combination that includes Carousel will have shorter max wait time than the rest. The configuration with the longest wait length becomes that of Ferris Wheel and Bumper Cars as these two types both attract the majority of groups arriving, creating long queues and longer max wait time. While Roller Coaster may have the longest queue length, when visitors arrive they may choose to join queues of other rides that they can take, thus reducing the number of riders on Roller Coaster queue, decreasing the max wait time.

Figure 12. Max Wait Time for Two Type Park

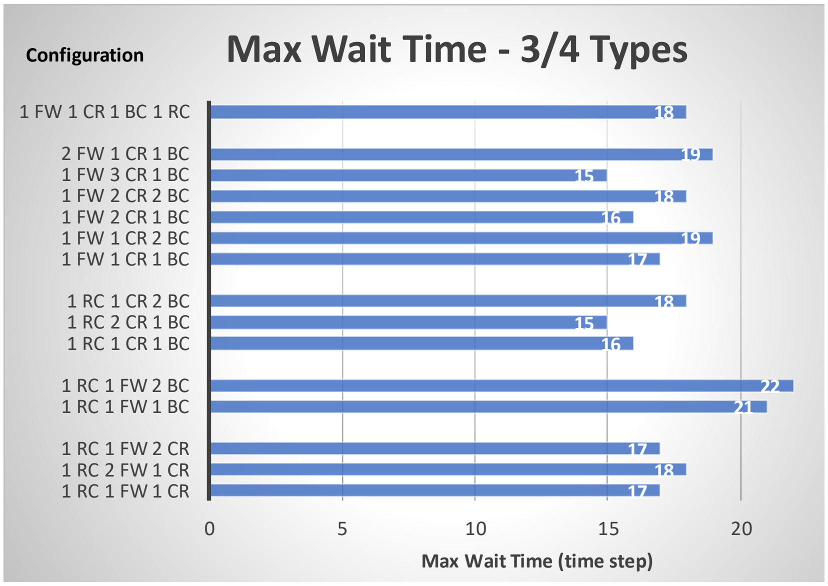


Figure 13. Max Wait Time for Three Type

As discussed above, when park has multiple rides, ride length begins to be a major factor. Similar to data in figure 12, in figure 13, any configurations which include Ferris Wheel and Bumper Cars, two types with the longest ride length, has the longest max wait time. The long ride length requires riders to wait more, which increases queue time and consequently the max wait time. The max wait time also begins to be consistent across all configurations, as queues now have many more guests, with fewer groups not being to find any rides suitable for them.

In general, the max wait time will correspond to the queue length, ride length and attracted groups of the configuration.

5. Conclusion

To minimize number of guests left early, the best configurations involves Roller Coasters and/or Carousels. Groups can only take either one of these two rides so adding any rides of either types provides more space for guests to join queues but do not increase the number of required rides for approximately half the number of groups.

If profit is the main concern, a configuration of only Ferris Wheels gives the highest return. This is, however, not very feasible in reality since guests will prefer a variety of rides. If this is the case, any combination that combines Ferris Wheel, which attracts all groups, with other ride types with different appeal groups, for example Roller Coaster and Carousel, is likely to do well. However, we have to take into consideration the assumption that visitor pays a certain price for each time they join a queue. If this is not true and visitors only pay once for entry, the increase in revenue for each addition of rides will be substantially smaller.

For minimal average wait time, it is crucial to choose a configuration with rides of shorter lengths, such as Roller Coaster. These rides will finish the rides quickly, clear the queues quickly and thus have shorter average time.

For max wait time, aside from ride length, other factors such as queue length and appeal groups also have to be taken into consideration. Also, as the number of rides increase, the max wait time will tend to stay consistent as there are fewer groups which will not be able to find any rides. Hence, there will a constant number of riders waiting in the queue.

6. References

Chun Wai Liew. *CS 150: Project I - Amusement Park Configuration.* (2018). Lafayette College.

Generate random numbers. (n.d.). Retrieved March 12, 2018, from http://www.javapractices.com/topic/TopicAction.do?Id=62

Oracle. Class ArrayList, *Java™ Platform*, Standard Edition 8 API Specification. (2017). Retrieved March 12, 2018, from https://docs.oracle.com/javase/7/docs/api/java/util/ArrayList.html

Oracle. Class HashMap, *Java™ Platform*, Standard Edition 8 API Specification. (2017). Retrieved February 26, 2018, from https://docs.oracle.com/javase/7/docs/api/java/util/HashMap.html

Oracle. Class Random, *Java™ Platform*, Standard Edition 8 API Specification. (2017). Retrieved March 12, 2018, from https://docs.oracle.com/javase/7/docs/api/java/util/Random.html