You Must Be At Least This Tall To Ride This Paper

Control 27

Contents

List of Figures

1 Introduction

A queue is a universal headache. It is never fun to be stuck standing in line. As most people do not spend a majority of their day caught in a queue, we have a tendency to accept its presence in modern living. Being caught in a queue will lower one's perceived experience of their surroundings, and can have a negative impact on an establishment's image, and eventually business.

Amusement Parks, while a favorite vacation destination, have acquired a debilitating association with endless lines. We are presented with the problem of implementing alternative ticketing schemes, called QuickPass Schemes, to cut down lines and maximize enjoyment of guests at an amusement park.

We develop a model utilizing probabilistic processes to simulate the movement of guests in an amusement park, with each ride varying in popularity. A variety of potential QuickPass ticketing schemes are developed and integrated into our model of an amusement park. We define the enjoyment of our park to be a simple mathematical relation, and study the effects of each QuickPass scheme on the population's enjoyment of our park.

We find that a scheme where QuickPasses act as placeholders in a ride queue creates the highest increase in enjoyment by our definition. This frees up the rider to do more in the park and still come back to enjoy that ride with a much shorter time in line.

Since reliable data on amusement parks is difficult to find, our model may not be as realistic as we would like.

2 Basic Model

2.1 Definitions

- Enjoyment is the following ratio: $\frac{\text{Average Number of Rides Per Person}}{\text{Average Time Spent in Line}}$
- A QuickPass is a ticket with a return time: a time interval. If a person has a QuickPass, they are authorized to return to a ride during that time interval to gain access to the special QuickPass queue.
- A QuickPass is *expired* if the current time is later than the return time.
- A QuickPass is *redeemed* when a person holding it enters the QuickPass queue.
- A QuickPass is *active* if it is not expired and has not been redeemed.
- A QuickPass is *live* if the current time lies within the return time.

2.2 Commonly Used Variables and Expressions

- t is the current time.
- L_t is the current number of people in the all queues at a specified ride at time t.
- Q_t is the current number of active QuickPasses for a specified ride at time t.
- ω_t is the estimated wait time at a specified ride at time t. ω_t is ALWAYS the sum of the people in line and the number of active QuickPasses. Alternately;

$$\omega_t := L_t + Q_t$$

2.3 Model Development

2.3.1 Park Design

We begin by creating a model amusement park to test potential QuickPass schemes. Our idealized park is open daily for 840 minutes (14 hours). Our park contains three types of attractions:

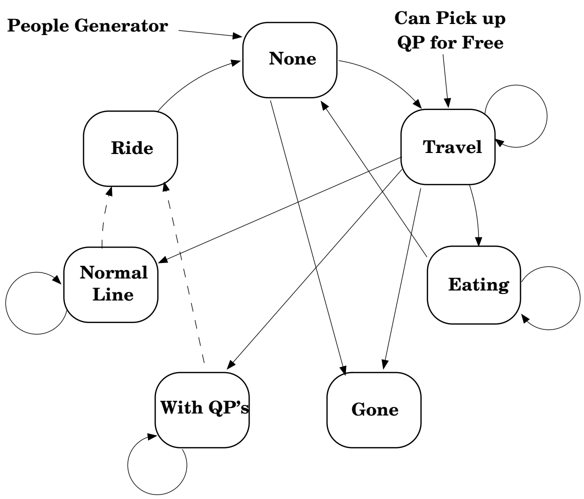

Figure 1: Visual Representation of Amusement Park Model with the implementation of the QuickPass System

- Big Rides: Designed to represent major attractions in an amusement park. Develops queues during busy periods of the day. Each Big Ride is capable of having two queues, a normal queue that any patron can join, and a QuickPass queue that only patrons with an active QuickPass can join.
- Small Rides: Designed to represent smaller attractions in an amusement park, including less popular rides (e.g., teacups). Develops queues during busy periods of the day.
- Food Vendors: Designed to represent restaurants in an amusement park. Traps the potential rider duration for a period of time.

Our model amusement park assumes all activities are equidistant from each other. In an architectural tradition that dates back to the late 1800s, amusement parks have been laid out into "Spacial Zones", in which each zone has its own unique theme and offerings.[1] Grouping a few major attraction per zone with correspondingly more smaller attractions is also a well known pattern in amusement park design [1]. Our model tests potential QuickPass schemes in one zone of the park. Assuming the traffic in each spatial zone is relatively consistent, our model can easily be applied to approximate the effects on a larger, multi-zone park.

2.3.2 Rider Characteristics

Our model has a simulated population of *automated state machines* that "ride" attractions. The population grows based on a generator function that shall be discussed later. The model simulates a population wandering around the park, searching for entertainment and food.

The people in our park cycle through a variety of states, illustrated in Figure 1.

- None: When a person enters the park, or completes a ride, they begin in the None State, and immediately move into the "Travel" state.
- Travel: A person in the "Travel" state is searching for something to do. They will approach one of the attractions, which are weighted to provide a sense of popularity. Big Rides will draw a larger crowd than Small Rides, and so forth. *If a person returns to the "Travel" state with an active QuickPass, s/he will automatically go join the QuickPass queue.*
- Eating: When a patron chooses to approach a restaurant, s/he is held in the Eating stage for a fixed period of time to correspond to the time that would be allotted to eating. Afterwards, the patron returns to the "Travel" state.
- Gone: Our simulated people can also leave the park. It is assumed that the average amusement park patron spends 8 hours in the park. A person can choose to move from the "Travel" state to the "Gone"

state; removing them from the population. After a fixed time, representing the closing time, all patrons will move to the Gone state.

When a person approaches a ride, s/he will go through an algorithm to determine his/her actions:

- First, a person can potentially request a QuickPass if they approach a Big Ride. Data from an amusement park that is currently operating with a similar system to a QuickPass system shows that 80% of park patrons are using it. [2] Therefore, we set our patrons to request a QuickPass 80% of the time.

  Next, the person will decide whether or not to join the queue for the ride.

- Refuses Normal Line: When faced with a long line, some people will choose not to join the queue and is forced to make into the travel. This is termed *balk* in terms of queuing theory.

- Joins Normal Line: The person joins the queue at the end of the line by moving to the "Normal Line" state. It is assumed that no people will leave the queue.

2.3.3 Modeling Park Population

While the main concern is behavior of our park while it is heavily populated, it would be naive to assume that the park begins a typical day with a large population. Our simulation contains a people generator which acts as the entrance to our theme park. Since a decline in population is the result of the simulated people choosing to leave, we are left to model the people as they arrive at our park.

Unfortunately, we were unable to find published data on daily admission patterns for an amusement park, and even the data on maximum capacities for amusement parks. This is rightfully so due to the privacy policy parks are required to adhere by.

Due to this difficulty, a generalization argument has to be made. Since our main objective is to stress test our model with a maximum population in order to test the efficiency for a testing scheme. Thus, we use the following impulse function that cuts off after the park has been open for 360 minutes (6 hours):

$$\Delta Population = \begin{cases} \frac{(PeakPopulation)}{(360minutes)} & \text{if } t \leq 6 \text{ hours;} \\ 0 & \text{if } t > 6 \text{ hours.} \end{cases}$$

People are added to our park on a constant basis until we hit a predefined maximum. This population growth equation is kept constant throughout all our schemes and therefore we are not required to mimic the potentially complex and unpredictable true-to-life guest arrivals in order to provide fairly exact outputs.

2.4 Model Strengths

- Stability: Consistent population inputs produce consistent results.
- Simplicity: There only exist two interacting parts that interact in predictable ways. Simple counting people, and integrating different QuickPass Algorithms.

2.5 Model Weaknesses

- Relatively Little Supporting Data: We were unable to find data to use as a basis for comparison, and justification. The model was developed by assuming, justifying, and building on top of each other, logical arguments to reach a purportedly valid model.
- Consistent Line Upper Bounds: Each potential Rider in the system will wait just as long for a popular ride as they will for an unpopular ride. This is something that could easily be cleaned with true-to-life data, but for now it just happens to be a busy day at the teacups.
- Expired QuickPasses: When patrons take a QuickPass, their return time is not part of the consideration. Potentially large numbers of people can leave the park without redeeming their QuickPasses.

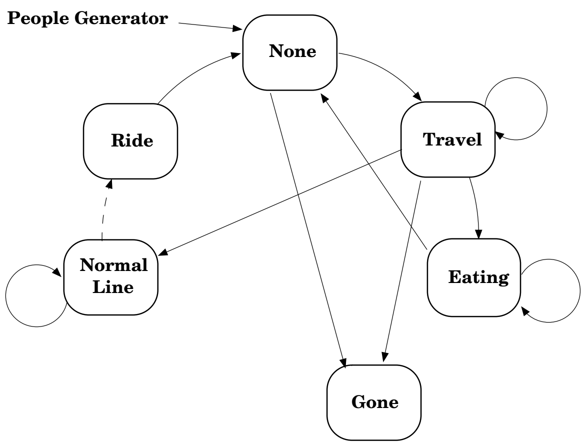

Figure 2: Visual Representation of Amusement Park Model by setting QuickPass distribution to zero

Most, if not all, of these problems can be fixed in the future versions of our model. However, all that we need is the data to input into our model. Since this is currently unavailable, me take our resultant assumption as the best case we can deal with.

3 Park Operations Under Various QuickPass Schemes

3.1 The Control

In order to have something to compare against, all tests were first run with no QuickPasses. Although the QuickPass algorithm is already built into the model, all that is necessary to make our model run without them is by setting the total number of QuickPasses distributed to zero.

3.2 A Crude QuickPass Scheme: Fixed Interval Rule and Variable Interval Rule

For comparison with the control, we consider a scheme which establishes no restrictions on the number of QuickPasses that are issued, called the Fixed Interval Rule. Calculating ω_t only at the beginning of a fixed interval of length λ, (15 minutes in our simulation) all Riders requesting QuickPasses receive a return interval of:

$$I_{return} = (t + \omega_t, t + \omega_t + \lambda)$$

Upon returning to redeem their QuickPasses, the Riders are integrated with the Normal line on a proportional basis. Riders are allowed onto the ride with one goal in mind: Forcing the QuickPass queue and the Normal queue to return to zero at the same time. Any borderline case is given to the QuickPass Queue to encourage its use.

While this scheme should work fine for smaller loads on the QuickPass system, high loads present a problem. Returning QuickPass users are subject getting through the interval irrelevant of how many people are in that interval. Since rides have a max speed, they can only handle so many people per interval. Thus there exists a possibility in this scheme where people are not serviced. We conclude, therefore, to see a select few QuickPass riders being "stuck" in line for long periods of time.

A simple variation on this scheme, the Variable Interval Rule recalculates ω_t during a shorter interval, and creates much larger return intervals (λ) for issued QuickPasses (1 hour in our testing). The desired effect of this system would be that QuickPass users would have an even larger interval window to arrive at the ride without sacrificing service quality. However, because it builds on the Fixed Interval Rule which is already expected to have an undesirable outcome.

This scheme is expected to fail. There are unbounded numbers of park patrons getting Quick Passes for intervals of time where only a fixed number of people can ride the Big Ride. It is expected that both queues to get unreasonably long, and thus patrons spend as much, if not more time in queues in the control model with fewer rides.

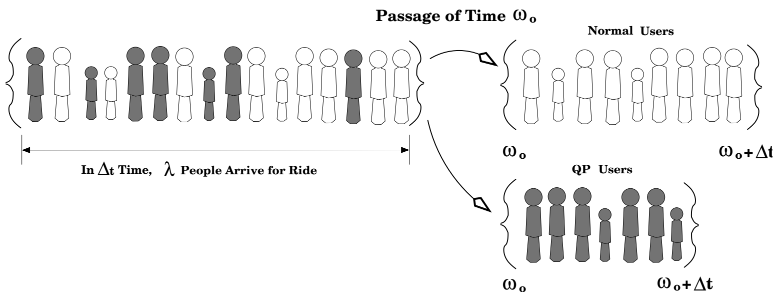

Figure 3: Depicts the principle behind the Fixed Interval Rule, which is specifically developed to preserves order in the queue. In the interval of time Δt, anyone that arrives at the QuickPass distributor, is given a time interval to return that is the same interval as if they never left the line. The QuickPass return times that are given by ω_0 to $\omega_0 + \Delta t$, where ω_0 is the projected wait time for the first person in the interval of time and Δt is the amount of time QuickPasses that are to be handed out within the same interval of time.

3.3 The Shortest QuickPass Line: A Bounded Scheme

The goal with this proposed scheme is to place a bound on the number of people on the ride at any given time that board using a QuickPass. This is accomplished by setting a fixed number of QuickPasses that can be given out for any return time. QuickPasses will be distributed with the closest return time that is not at capacity.

Assuming the QuickPass return times are distinct time intervals (NO overlap), it is trivial to figure out how many QuickPasses to distribute for each return time. It is not much more difficult to decide how many passes should be given out during overlapping intervals. Taking the number of passes that would be distributed per distinct time interval, say N, and assuming that K QuickPass return times would be overlapping at any time t, the number of QuickPasses that should be distributed for each return time N_k is:

$$N_k = \frac{N}{K}$$

If patrons carrying live QuickPasses returned to the queue uniformly (which we do not assume), will result in N_k people in the QuickPass queue at in the overlapping period, the exact number that will board the ride during that time.

This results in the QuickPass queue having a reasonable upper bound at all times, creating an increase in enjoyment for those patrons with a QuickPass. However, the number of QuickPasses distributed in a day will be small compared to the other schemes we consider. The park will run out of QuickPasses before most people can take advantage of them. When wait times and total rides are averaged throughout the park, this model is expected to succeed with rough success.

3.4 A Meta-Marker: The Virtual PlaceHolder Rule

In a perfect world, a bored rider could merely ask someone else to hold their place in line. It is possible to implement a system that does exactly that using QuickPasses. By counting the number of people entering the Normal Queue and the active QuickPasses, we can determine when a new QuickPass user will reach the front of the line if s/he joins the queue. This concept is more accurately illustrated in Figure 4. The return interval, of length λ is set to force the QuickPass holder back to the queue in this manner:

$$I_{return} = (t + \omega_t - \lambda - \delta t, t + \omega_t - \delta t)$$

A returning QuickPass holder will arrive back to the queue δt before ω_t. By treating each QuickPass as a placeholder, when a one reaches the front of the queue, we take the first person in the QuickPass queue and place them on the ride.

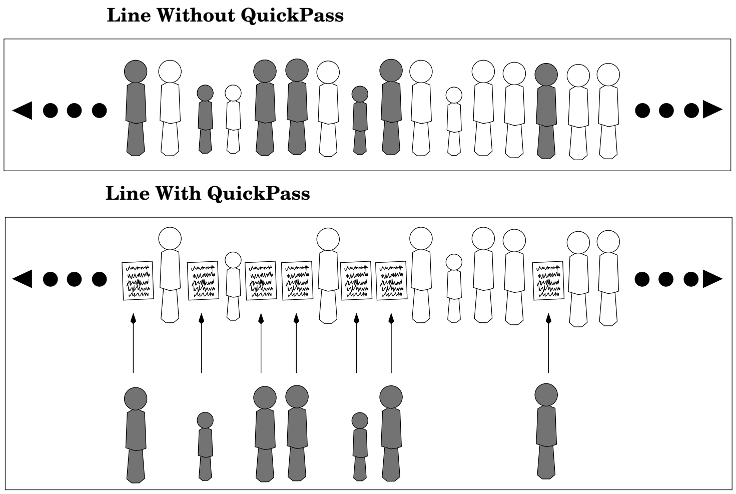

Figure 4: Depicts the role of the Virtual Placeholder Rule that is specifically designed to keep order in the line and bound the total number in the line. The people shaded gray represent the people that would chose the QuickPass option if it is available. Notice that the QuickPass ticket is metaphorically left in the line to signify the position reserved by the QuickPass ticket.

4 Analysis of Results

4.1 Results from Schemes

4.1.1 Does Our Data Really Mean Anything?

The following data was collected as averages of over 100 simulations at two population setting. The Each value is calculated as follows:

$$\omega = \frac{1}{100}\Sigma(AverageDailyWaitTime)$$

$$r = \frac{1}{100}\Sigma(AverageDailyRidesTotal)$$

$$\rho = \frac{\omega}{r}$$

$$E = \frac{r}{\omega}$$

All times are in minutes, and the final column is the amount of time the average park patron spent on rides each day.

Observing the values in the table below, we see that regardless of a heavy or light test, the maximum and minimum of the r value is less than one. Most people do not get to ride any additional rides! If our total calculation for enjoyment was solely based on rides, all of our scenarios would be failures. However, the time spent waiting in line varies enough to change our enjoyment values, making our results worth studying.

Additionally, no significant standard deviations were found on any test, so our simulation is stable enough to use.

4.2 Numerical Results

		No QuickPasses	Fixed Interval Scheme	Modified Interval Scheme	Virtual Placeholder Scheme	Bounded Scheme
Low Congestion	Wait time ω	206.068506	183.9429615	199.044313	149.1686935	164.7552175
	Rides r	7.8413814	7.6820908	7.4651014	8.2205856	8.4645683
	Time on Rides ρ	26.279618	23.9443875	26.66331	18.1457505	19.4641015
	Enjoyment E	0.1902615	0.2088172	0.1875236	0.27554661	0.2568832
High Congestion	Wait time	268.8796975	190.319273	195.08904495	168.633396	182.643352
	Rides r	4.4203402	4.0690933	3.9766510	4.1362875	4.2525484
	Time On Rides ρ	60.827829	46.7719115	49.058628	40.769264	42.9491535
	Enjoyment E	0.0821992	0.1069018	0.1019189	0.1226414	0.1164167

To our surprise, all of our QuickPass schemes yielded results superior to our control. While this was somewhat unexpected, each set of results must be analyzed against some of the weaknesses of the schemes and our definition of enjoyment.

The Fixed and Variable Interval Schemes were both expected to fail. However, they both yield higher enjoyment levels than the control. As previously discussed, some unlucky Riders were expected to be in the QuickPass line for long periods of time. It would appear that this set of Riders is insignificant in the final calculations, but these long QuickPass lines still do exist, and we consider them to be unreasonable result in choosing a successful scheme. Further, the prediction that the Variable Interval Rule would have less success than the Fixed Interval Rule held. That means that the errors in the Fixed Interval Rule carried over into the Variable Interval Rule and were further magnified by the current errors.

The Virtual Placeholder scheme performs well against our control and the other tested schemes. This scheme is our overall best scheme and performs best when the park is at or near peak capacity, yielding an average queue decrease of 20 minutes per ride.

The Bounded Scheme also does well in our simulation but performs significantly better on peak capacity days. The variance in performance based on park population is fairly predictable because this scheme reduces the average time in line for a QuickPass user as compared with the population of non-users. On a non-peak day, when the average time in queues for a non-user is lower, QuickPass users will have less of an advantage. It is also worth noting that time intervals are issued a minimum of three hours in advance. If the Bounded Solution had a minimum of zero, it may have performed better in comparison to Virtual Placeholder scheme.

5 Conclusion

Our task was to recommend a scheme for distributing QuickPasses that maximizes guest enjoyment. We have defined enjoyment as a simple ratio of average number of rides per person over the average amount of time spent in line. By using argument that build on each other to compensate form the lack of concrete data, we found that:

Using a Virtual Placeholder Scheme maximizes guest enjoyment.

We also note that a Bounded scheme works admirably and can be considered as an alternative should the Virtual Placeholder become less efficient than the Bounded scheme.

5.1 Possibilities for Future Development

- Test model with realistic data from justifiable sources One of our greatest disappointments was not being able to test our park with realistic arrival rates.
- Tweak Schemes Each existing scheme can be modified, particularly the Bounded Scheme.
- Improve Rider Algorithms to reflect ride popularity - less popular rides should usually have a shorter line.

- Test Additional Potential Schemes Can we merge two schemes? Test combinations of schemes? Come up with something new?
- Modify working schemes to try to represent the anomalies that were introduced in the question, thereby proving that our model fully corrects those anomalies.
- Explore probability that QuickPass user returns to QuickPass queue

References

[1] Loius Wasserman, *Merchandising Architecture*, Wasserman, 1978.

[2] Karen Auguston Field, *Design News*, Feature Article, Vol. 58, Iss.1, 2003