Figure 8 illustrates shows for a fast lane;
Figure 9 illustrates generation of voucher;
Figure 10 illustrates a pop-up notification for a voucher;
Figure 11 illustrates a wallet containing vouchers;
Figure 12 illustrates a simulated output of a crowd analysis and prediction module;
Figure 13 is a schematic block diagram of a mobile phone;
Figure 14 is a schematic block diagram of modules and sub-modules of a virtual queue app running on a mobile phone;
Figure 15 is a schematic block diagram of a virtual queue kiosk;
Figure 16 is a schematic block diagram of modules running on a virtual queue kiosk;
Figure 17 is a schematic block diagram of a virtual queue token terminal provided at an end of queuing area before entering an attraction;
Figure 18 is a schematic block diagram of a module running on a virtual queue token terminal;
Figure 19 is schematic block diagram of a computer system which can be used for a virtual queue management console;
Figure 20 is a schematic block diagram of modules running on a computer system used for a virtual queue management console;
Figure 21 is a schematic block diagram of a virtual queue slot prediction model;
Figure 22 schematically illustrates a method of dynamic pricing;
Figure 23 schematically illustrates a method of Bayesian optimization for prising scheme searching;
Figure 24 illustrates an example of an itinerary plan displayed on a mobile phone;
Figure 25 schematically illustrates an algorithm architecture for generating an itinerary plan;
Figure 26 schematically illustrates itinerary item selection;
Figure 27 is a schematic block diagram of multi-agent modelling and simulation system;
Figure 28 schematically illustrates searching for a sequence of attractions to be visited; and
Figure 29 illustrates process of using patron and attraction agent behavioural models.
Figures 29A and 29B illustrate process of using patron and attraction agent behavioural models according to Figure 29.

Detailed Description of Certain Embodiments

In the following, like parts are denoted with like reference numerals.

- there are fast pass holders and normal day pass holders, with a ratio of 1:9 (the ratio is tuneable)

Other scenarios may be used.

Rides:

- each ride operates according to designed timing information provided. Key information used includes: theoretical hourly ride capacity as the hourly throughput, dispatch capacity, dispatch interval, normal lane capacity, fast lane capacity, and admission ratio (fast: normal) = 4:1 (the ratio is tuneable);
- at the end of each (riding) cycle, the ride will clear the batch of patrons and take a new batch of them from the fast and normal lanes according to the empirical ratio of 4:1. If not enough patrons are waiting in the fast lane, the remaining will be taken from normal lane to fill the dispatch capacity;
- each ride will be initialized with a random popularity to attract patrons to visit (the popularity is tuneable); and
- VQ Tokens can be manually added to each ride to mimic the dynamic supply of VQ tokens in the VQ gamification system to attract patrons.

Patron behaviour:

- there are two types of patrons: hardcore (intense) and normal. For hardcore visitors, they target to play as many (different) rides as possible (the number of rides is tuneable);
- a patron visits one ride after another, with random resting time in-between, e.g., normal distribution around 15 mins (the rest time is tuneable);
- the choice of the next ride is based on multiple factors, which currently include: ride popularity, distance from the current location, estimated waiting time of the ride, and the VQ Tokens to be earned;
- the longer the waiting time, the less “resting and shopping” time from the interval of rides.

Different patron profiles can be reflected by using different parameter configurations. For example, model parameters θ_1, θ_2, θ_3 (Figure 29B) can be varied.

Even with a simplified scenario and model, the simulated results reflect the complicated interaction among different entities in real-time.

to make best use of the limited available space available, such as queues of different attractions, and divert patrons more across the park more evenly to different locations based on the current situation.

Data sensing and understanding patron distribution is important in trying to understand the current situation. Even with sensors deployed with the VQ solution such as the VQ validation terminals 25 (Figure 2) and the VQ token collection terminals 26 (Figure 2) at selected locations, footprints of patrons can only be obtained at a coarse level of granularity both in space and time. Herein, multiagent modelling is used to improve the level of granularity. The objective is to provide higher granularity of crowd distribution information through simulating patron's next attraction choice behaviours based on existing data, such as sensed density of localities, partial footprints of patrons, operation status of attractions and VQ functions utilized by patrons through the provided VQ app 22 (Figure 2), such as itinerary planning and incentive utilization.

Furthermore, key parameters regarding patron behavioural choices in the simulation model is calibrated using evolutionary algorithm approach, which can be achieved with collected observed crowd count data at selected locations along time in the theme park.

Referring to Figure 28, an agent-based modelling and simulation system 235 is shown. The system 235 models patrons and attractions as autonomous agents and simulates their behaviour and interaction over time.

The system 235 generally includes patron agents 236 and attraction agents 237 which define a model 238 and a scheduler 239 which controls agent activation and handles time in the model.

Referring also to Figures 29, 29A, and 29B, the patron agent 236 include a patron decision making module 241 for deciding a next attraction choice and a patron behavioural module 242 for executing a current choice, and the attraction agent 237 includes an attraction behaviour module 243 for interacting with patrons.

Upon arrival of the theme park and after binding their tickets into the VQ app 23 (Figure 2) (step S24.1), the patron interacts with the VQ system. Decision logic of the patron agent 236 determines whether the patron has "Next Attraction Choice?" (step S24.2) to identify along which branch further computation will proceed.

If the decision is "No", then patron agent 236 executes the patron decision making module 241 to decide next attraction choice. Otherwise, the patron agent 236 executes the patron behavioural module 242 to execute a current decision to simulate the process of arriving at an attraction, joining a queue, waiting in the queue, enjoying the ride and leaving the attraction. The process also depends on the interaction with the attraction agent 237.

Patron decision making module

The decision-making process of a patron agent is based on the status of the agent with regard to the VQ functions. For example, they can make an advance reservation of VQ slots for an attraction to join the fast lane at corresponding time slot. Additionally or alternatively, they can use the itinerary plan function 94 (Figure 14) of the VQ app 22 (Figure 13) to obtain guided itineraries, where they secure incentives upon completion. Additionally or alternatively, they can get time-constrained vouchers to redeem at specific facilities, such as restaurants, as incentives to encourage them to visit. The decision-making process of a patron agent is also based on user profile.

Referring to Figure 29B, the decision process checks whether patron agent "Has Immediate Reservation" in the following simulation time period T_1', which is a period defining how near the upcoming reservation is from the current simulation time. If "yes", then based on a model parameter θ_1, the algorithm determines whether the patron agent 236 has an upcoming reservation (or "immediate reservation"). Here, θ_1 represents a tendency of a patron to commit to their advance reservation and can be tuned computationally or calibrated through data collection from the actual patrons' footprint in practice. If "no", then the algorithm proceeds to the next decision logic whether the patron "Has Itinerary Plan?"

The algorithm checks whether the patron agent 236 "Has Itinerary Plan". If "yes", then, based on a model parameter θ_2, the algorithm determines whether the patron agent 236 chooses the next attraction on the itinerary plan as the next attraction choice. Here, θ_2 represents a tendency of the patron to commit to their itinerary plan and can be tuned computationally or calibrated through data collection from the actual patrons' footprint in practice. If "no", then the algorithm proceeds to the next decision logic whether the patron "Has Time-Constrained Voucher Redemption?"

In step PE2, the patron agent waits in a queue and the module update its sequence in the queue at simulation time t_1.

In step PE3, the patron agent leaves the queue and enjoy the ride as one of the batches of patrons to be loaded into the ride at simulation time t_2.

Finally, in step PE4, the patron agent leaves the attraction after taking the ride and update patron agent's current attraction choice as none at simulation time t_3.

The algorithm proceeds to the next iteration of simulation to execute the patron decision making process.

Attraction Behavioural module

Referring ~~still~~ to Figure 29A, an attraction at a theme park can be seen to serve patrons through four main processes:

In step A1, the attraction agent admits the patron ("queue admission"). At simulation time t_0', the attraction agent checks the VQ slot validity of the visiting patrons. If a patron agent has a VQ slot at the corresponding time, the patron agent is added into the fast lane. An assumption is made in the simulation that the fast lane will not be physically full due to the fact that the fast lane is mainly used to serve VQ slot holders and VIPs. The solution will moderate the total number of VQ slot holders during a slot period to prevent over-crowdedness. With the uncertainty of VIP group to join the fast lane, operation can still manoeuvre the output throughput ratio between fast lane and normal lane to prioritize the fast lane guests with shorter waiting time in practice. Thus, an assumption is made to simplify the queue admission process in the simulation model. For guests without a valid VQ slot at time t_0', a normal lane can admit guests if the normal lane has capacity, *i.e.*, is not full. Otherwise, the patron agent returns to the initial status of waiting to be admitted into queues.

In step A2, the attraction agent accommodates guests in queues ("host guests in queues"). During simulation period T_1', the attraction agent checks the simulation time to determine "Is Next Ride Cycle Ready?". *T1'* can be determined based on the ride's dispatch time, and full cycle time, which is further defined as the sum of the dispatch interval, ride cycle and loading and unloading time, which can be obtained from the operational fact sheets as scheduled. If a new ride cycle arrives at simulation time t_1' (=