Optimizing Parking Lot Performance: A Comparative Analysis of Allocation Strategies

Marcos Costa, Rodrigo Moucho and Rodrigo Póvoa
Department of Informatics Engineering
Faculty of Engineering, University of Porto
Porto, Portugal
{up202108869, up202108855, up202108890}@fe.up.pt

***Abstract*—Parking management, especially in urban areas of high demand, poses a challenge for highly populated cities, such as Porto and Lisbon, which is caused by the scarcity of the resources and the stochastic nature of the driver's behavior. This project aims to understand the parking allocation problem and solve it by modeling the interactions between drivers and the infrastructure to evaluate the different management strategies. It implements an Agent-Based Model (ABM) developed in Python [3] language using the MESA [7] framework. Drivers are deployed as autonomous agents with their own state machines, ranging from queuing to parking and exiting. The program evaluates multiple scenarios, including a standard First Come First Serve (FCFS), reservation based approach and dynamic pricing approach that adapts the charge rate based on factors such as the occupancy and driver willingness to pay. From each simulation, the model generates values for key performance indicators (KPI), such as occupancy rates, queue times, revenue and among others. The main findings of this project are only to provide information and to identify which policies were the most effective.**

***Index Terms*—Urban Parking Management, Agent-Based Modeling (ABM), Discrete-Event Simulation, Dynamic Pricing, Resource Allocation, Reservation Based**

I. Introduction

Urban parking management, or any other high demand parking area, is a problem that affects nearly every country in the world, especially very large and busy cities. The lack of parking spaces, inefficient parking management systems, and the growing demand are some of the main causes of this problem. By modeling drivers as agents of the system and parking spaces as resources to be managed, our simulation will allow the group to test different strategies and solutions to optimize a parking system. This project falls under the Field of Modeling and Simulation of Transportation Systems, particularly the management and optimization of parking.

The main issue addressed here is the Parking Allocation Problem, defined by the following statement: what is the best way to distribute a finite amount of parking resources to a stochastic stream of drivers to maximize both the resources and the revenue. This problem can be defined by the interaction between: resources, which is a set of parking spaces; demand, which is a stochastic arrival process of drivers with a intensity defined by a non-homogeneous poisson process, where each driver has a willingness to pay, a specific type and a dwell time. This optimization problem tries to address to the minimization of the waiting times and balking and the maximization of the revenue and occupancy rate.

Regarding the motivation, our main interest for choosing this topic came down to the fact that it is becoming a serious problem in densely populated cities of developed countries. As the urbanization intensifies over the years, the disparity between vehicle ownership rates and limited spatial resources has been slowly, but surely, transforming the parking into a major bottleneck for urban mobility. Testing new strategies in real world is extremely risky and expensive, which is why this project is important to provide an ability to test these new strategies without any risks.

The main target of this project is to develop a program that is able to model and simulate the dynamic interactions between the drivers and the parking lots in order to find the optimal parking allocation strategies. The goals are to develop an agent based model in python using the MESA library to simulate the interactions that were mentioned above. Other goals are also marked down, such as implementing and simulating different allocation scenarios and monitor/understand the Key Performance Indicators. The main research questions are basically to understand the upside and downsides of each strategy based on a multitude of factors.

Based on the topic, model and the strategies implemented, the main hypotheses will be:

- Hypothesis 1: Dynamic Pricing strategy will maximize the revenue but at the exchange of a higher balking rates, due to the charge rate being higher than the driver's willingness to pay (WTP).
- Hypothesis 2: Reservation strategy will certainly reduce the occupancy rate, but the revenue that comes from the reservation price will pay off that trade-off.

This report will respect the following format: Related Work, in which a literature review is done, Materials and Methods, where the problem is properly formalized, and Results and Discussion, which mentions the main results and insights of the metrics. To finish, we have the conclusions and future work.

II. Related Work

This section will assess the current state of the art regarding parking management strategies and simulation methodologies that are used to judge these approaches. A light literature review was done to highlight the current limitations of approaches.

A. Parking Policy and Smart Mobility

The concept of the urban parking as a problem has changed from being static to a dynamic management challenge. In the past, parking was viewed as a resource issue that could be solved by increasing the physical capacity of the parking lots. However, this is not true nowadays, due to the intensification of the urbanization, which makes expanding these infrastructures spatially/economically unviable.

The famous literature ”The High Cost of Free Parking” [1], which mentions the main issue which is that most of the minimum parking requirements are commonly arbitrary, either based fake statistics or bias from other places. These issues lead to an unnecessary amount of parking lot infrastructures which has negative impacts on the economy, such as increasing the house prices, and it also negatively impacts the environment by increasing the traffic congestion, which leads to more carbon emissions.

To solve the issue above, some modern approaches have started to focus on **Smart Mobility**, such as the European Comission’s ”Park4SUMP” [2] project, which operates in 14 European countries and more than 300 cities in Europe. The main takeaways of this project is to raise awareness and gaining acceptance among countries on how the right parking policies can help cities develop towards a better future, along with helping with building, specially in cities that have difficulties to do so. One of their documents ”PUSH & PULL: 16 Good Reasons for Parking Management” [2] , mentions that one of their core philosophies is that parking management works the best when both push (forcing the drives to stop using their cars and start using public transportation) and pull (baiting them into different alternatives using the money generated from the parking) measures. It is mentioned that the best occupancy rate should be around 85% to ensure that at least few parking spaces are free, therefore eliminating some of the drivers that do unnecessary laps around blocks in hopes to find a free spot.

B. Simulation Methodologies: Applications vs Libraries

The selection of a simulation tool is one of the most important and critical steps for the implementation of an Agent Based Model. The current tools range from specialized applications to flexible libraries integrated in popular programming languages.

NetLogo [4] and SUMO [5] are applications that are capable of simulating the scenarios of this report. The NetLogo is favored for its low learning curve and built-in visualization tools, however it has a big downside, which is the fact that it is limited by its programming language, ”Logo”, which doesn’t allow it to integrate with other data science pipelines, such as python’s Pandas. SUMO is actually the industry standard, due to its robustness and small traffic simulator capable of modeling complex road infrastructures and car interactions. The main downside of using SUMO is the fact that it is highly overkill for a small project like this one and it requires much more experience to implement internal logic, unlike other alternatives.

Regarding the domain within Python libraries, there are two that standout: SimPy [6] and Mesa [7]. SimPy is a process based discrete event simulation framework based on standard Python, which makes it excellent choice for modeling asynchronous processes, such as queuing systems, but it does not have support for spatial grids or agent based interactions. On the other side, Mesa is an agent based modeling framework that was inspired by NetLogo and it allows for the creation of a multitude of objects, such as agents, spatial grids (for parking lots) and schedulers, while at the same time integrating with other Python libraries for an even deeper analysis of the results that resulted from the simulations.

It is possible to see a Table I

TABLE I
COMPARISON BETWEEN SIMULATION FRAMEWORKS

Tool	Type	Positives	Negatives
NetLogo	Application	Rapid prototyping, built-in visualization and low level learning curve.	Uses Logo language and little integration with external tools.
SUMO	Application	Industry standard and realistic large road infrastructures.	Overkill for a small project.
SimPy	Python Library	Excellent for queuing logic and its lightweight and fast.	Does not have spatial/grid support and no visualization.
Mesa	Python Library	Native Python integration and modular, flexible data collection.	Slower execution than C++ and visualization requires browser setup.

C. Research Gap

After the literature review and a short research, it was observed that there is little to none integration of advanced AI models, such as Deep Learning and Reinforcement Learning, which makes it a worthy field to explore in the future. However that domain will not be within the scope of this project, due to its complexity.

The alternative is that this project will focus on exploring the deterministic evaluation of allocation rules, such as the already mentioned strategies. There is a lack of studies that actually isolate each strategies and compare these policies within a controlled, but stochastic agent environment.

III. MATERIALS AND METHODS

This section introduces the formal definition of the chosen topic, optimizing parking lot performance and comparing strategy performances, along with the tools used and the architecture of the Agent Based Model developed to simulate the interactions between the drivers and the infrastructure.

A. Problem Formalization

The simulation is modeled as a Discrete Event system (DES) employing on a rectangular grid G with $W \times H$ dimensions. The primary aim of this simulation is, like mentioned before, analyzing the main upsides and downsides of each strategy, taking into consideration the minimization of the waiting times and balking and the maximization of the revenue and occupancy rate.

The parking lot P is defined as a finite amount of resources $R = \{r_1, r_2, ..., r_C\}$, where C is the total capacity of the parking lot. The demand is defined by a stochastic stream of drivers/agents A. The problem can be formalized as:

- **Arrival Logic:** Agents arrival is defined by a Poisson distribution with intensity $\lambda(t)$, where t represents the step and the aim with this approach is to represent peak and low activity hours.
- **Agent Properties:** All agents a_i, where i represents their id, have some properties to help with the simulation:

$$a_i = \{WTP_i, P_i, D_i, S_i\} \tag{1}$$

Where WTP represents Willingness to Pay (maximum price that they are willingly to pay), P is Patience (a small probability where drivers can decide not queue up if it is too long, D is Dwell time, and S is the current driver/agent state ($ARRIVING$, $APPROACHING\_GATE$, $WAITING\_AT\_GATE$, $DRIVING\_TO\_SPOT$, $PARKED$, $EXITING$, $EXITED$).
- **Constraints**: The simulation contains constraints, such as the gate service rates and the total capacity of the parking lot.
- **Objective:** Comparing the main upsides and downsides of each strategy, while looking at metrics such as R, which represents the rate ($/min) and the average occupancy.

B. Materials and Tools

The simulation environment is built in **Python 3.13.2**, and uses the following libraries:

- **Mesa:** A python library, which is a modular framework for Agent Based Modeling, widely used to handle agent interactions, creation of spatial grids and continuous evolution of the simulation.
- **Pandas & NumPy:** Integrated within Mesa's framework and it is used for for data collection and statistical analysis.
- **Matplotlib:** Integrated within Mesa's framework and is used for live visualization of occupancy trends and queue dynamics.

The simulation runs on a default discrete time-to-step basis, where 1 step corresponds to 1 minute of real world time flow.

C. Methods: Data Generation and distributions

Since there was not any free and public data to help us develop a statistical function to distribute the behavior of the drives, we decided to come up with our own uniform distribution, as seen below.

- **Dwell Times:** It is implemented a uniform distribution to represent diverse parking needs, to properly represent agent behaviors:
  - *Short Stays (15%):* Uniform distribution $U(40, 140)$ minutes (quick errands).
  - *Normal Stays (60%):* Uniform distribution $U(240, 300)$ minutes (work/meetings).
  - *Long Stays (25%):* Uniform distribution $U(300, 500)$ minutes. (long work shifts).
- **Willingness to Pay (WTP):** Defined as the log-normal with 2 distinct parameters, as seen below.

$$WTP_i \sim \text{Log-Normal}(\mu, \sigma) \tag{2}$$

The parameter are set at $\mu = \ln(0.044)$ and $\sigma = 0.34$, which causes a log normal distribution where the average driver rates the parking lot service at around 0.044 €/min.

D. Methods: Agent Flow and State Machine

Each driver has their own logic, which is defined by a finite state machine (FSM). There is 8 different states, already defined above, and upon the instantiation of the agent/driver, it evaluates the environment:

1) **Price Verification:** If the $Current\_Price(t) > WTP_i$, the agent balks immediately.
2) **Congestion Check:** If the queue length is higher than 6 and the current occupancy rate is $> 80\%$, there's a chance, affected by the queue length and their willingness to wait, that the agent leaves.
3) **Service:** If accepted, the agent/driver enters the queue, waits in the queue, passes the gate and is given a spot coordinates (x, y) within the parking lot.
4) **Parking:** The agent/driver occupies the resource for duration D_i.
5) **Leaving:** After the agent/driver's dwell time, D_i, is up, they will start leaving the parking lot.

E. Experimental Scenarios

To evaluate the initial hypothesis defined and to actually start using the simulation model, three different management strategies were deployed:

1) **Baseline (FCFS):** A default price model where the parking spots are given accordingly to First Come First Serve basis. The only constraints here are the arrival logic and the number of available parking slots. The pricing rate is fixed at all times and all parking spaces are available to any and all agents. Regarding the behavior, agents only balk if the queue length is higher than their willingness to wait or if the price exceeds their willingness to pay. This scenario is viewed as the group control of this experiment, to better evaluate and compare with other strategies
2) **Reservation Based:** In this strategy, there is two distinct parking spaces: general spaces, defined as C_{gen}, and

VIP spaces, defined as C_{vip}. A small subset of the agents/drivers will be considered as reservation holders when they are generated and these drivers will be guaranteed entry in C_{vip} spaces, where they will go through a different gate, to avoid congestion. To add a touch of realism, the logic also introduces a "no show" probability, meaning agents that booked the reservation but never showed up and a "hold time", where a reserved spot will stay empty until the specific agent arrives, which will lower the utilization rate of the facility.

3) **Dynamic Pricing:**This approach implements a supply and demand logic pricing model to regulate the demand, and to maximally profit from said demand. The parking rate $R(t)$ is continuously calculated at each step based on the current occupancy ratio, defined by a linear surge function with an elasticity factor α:

$$R(t) = R_{base} \times (1 + \alpha \cdot O(t)) \tag{3}$$

Another crucial feature of this strategy is the fact that the dynamic price creates a interactive loop on the arrival logic. This means that while the raw generation of the agents still follows a Poisson process with intensity λ, the actual arrival rate, $\lambda_{eff}(t)$, becomes a function of price. The agent i that arrives at t will enter the queue if:

$$WTP_i \geq R(t) \tag{4}$$

A good example is at the start of the simulation, $O(t) \rightarrow 1$, $R(t)$ spikes, which causes $R(t) > WTP_i$ for a big part of the agents.

IV. Results and Discussion

A. Methodology of Data Collection

Given the stochastic type of the arrival logic and the probabilistic distribution of said agent attributes, such as Willingness to Pay and Dwell Time, each one of the three management strategies were executed for $t = 1000$ steps, representing 16 hours. The data from each one of these runs were harvested using the `Mesa DataCollector` module from MESA, which recorded the following

- **Traffic:** "Total Arrivals" across the entire runtime, which is a counter of agents that entered the parking lot (not counting reserved drivers) , and "Did Not Enter" are drivers that decided not to join a long queue plus drivers turned away because the price was higher than their willingness to pay.
- **Queue:** "Total Queue Time" is a sum of all drivers queue duration, in steps. "Queued Drivers", which is a counter of the drivers that queued and then entered. "Avg Queue Time" is the division between "Total Queue Time" and "Queued Drivers".
- **Reservations:** "Fulfilled" is the counter of reservations that were actually used. "Idle Reserved Spaces", which is the number of VIP spaces currently flagged as reserved but empty. "Not Fulfilled" is the number of reservations that were missed.
- **Financials:** "Rate" represents the rate at which the drivers are being charged with (during the dynamic pricing, this value changes over time). "Total Revenue" is the sum of all payments collected. "Lost (Price)" represents the number of drivers turned away because the current price exceeded their willingness to pay.
- **Occupancy:** "Current" which represents the percentage of the occupancy at a step t. "Average" is computed as the sum of "Current" at each step, divided by the total steps.

B. Performance Analysis

Starting with the baseline strategy, and by observing the Figure 1, we can observe that during the peak hours, there was a good amount of people waiting at the queue, due to the big influx of drivers at these peak hours. The number of drivers parked reached a peak of 120, which happened at the middle of the simulation run, which correctly resembles the real world statistics. From this strategy, 12 drivers balked away from a total of 246 drivers and the revenue was 1453 euros. The occupancy rate reached a peak of 60.6%. There isn't a lot of conclusions to be made from here as it this is the baseline and is used to compare against other strategies.

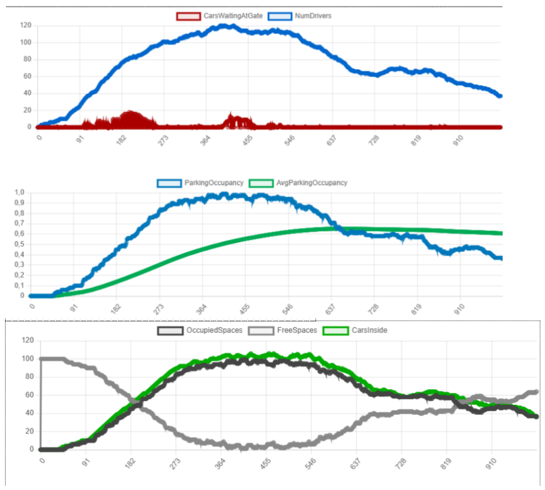
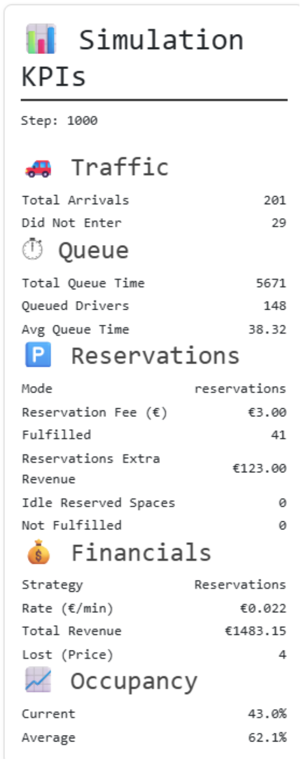


Fig. 1. Baseline Strategy Results.

Regarding the reservations strategy, two different runs with slightly different configurations. The first one, an additional lane for reservation entry only was made, and in the second one, there is only a single queue lane.

From the images, Figure 2 and Figure 3, we can observe that there was no difference at the income, however the average queue time of the reservations with 2 lanes is drastically less than with just one lane, which reflects into a lower number of drivers that balked away. The main takeaway from this is that with more reservations, there is less normal spots and "protected" reservations spots, which creates bigger queues. With one gate, people who reserve still have to stay in the line while with two gates, it yields the same revenue and the occupancy and queues decrease drastically. Although it yields more income than the baseline strategy and has higher average occupancy, it has higher average queue times.



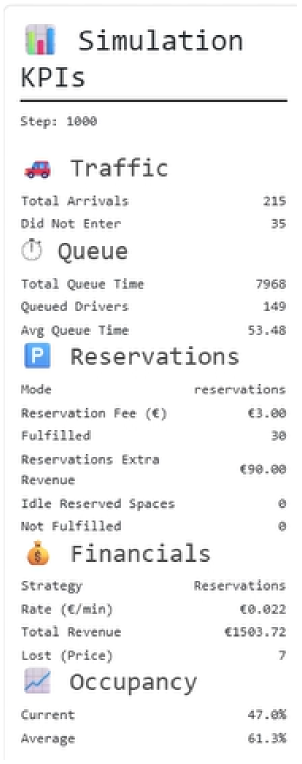
```
📊 Simulation
KPIs
Step: 1000

🚙 Traffic
Total Arrivals                 201
Did Not Enter                   29
⏱ Queue
Total Queue Time              5671
Queued Drivers                 148
Avg Queue Time               38.32
🅿 Reservations
Mode                  reservations
Reservation Fee (€)          €3.00
Fulfilled                       41
Reservations Extra
Revenue                    €123.00
Idle Reserved Spaces             0
Not Fulfilled                    0
💰 Financials
Strategy              Reservations
Rate (€/min)                €0.022
Total Revenue             €1483.15
Lost (Price)                     4
📈 Occupancy
Current                      43.0%
Average                      62.1%
```

Fig. 2. Reservation with 2 lanes.

Finally, regarding the final strategy, Dynamic Pricing, and similarly to the reservations approach, two different runs where in the first one, low prices increases slightly the demand, and the second one, where low prices drastically increases the demand. From the images, Figure 5 and Figure 4, we can observe that the run where the low prices greatly increase the demand yields way more returns than the other one, and, actually, on all and any metrics, it performs far better than the run where the demand only increases slightly. The second run offered us the best occupancy rate of all strategies altogether, but at the exchange of a higher number of drivers that balked away. It is safe to say that an extremely high queue times and rejected demand may discourage future use by clients and that there is a higher revenue for the second run, but it relies on attracting much bigger demand.

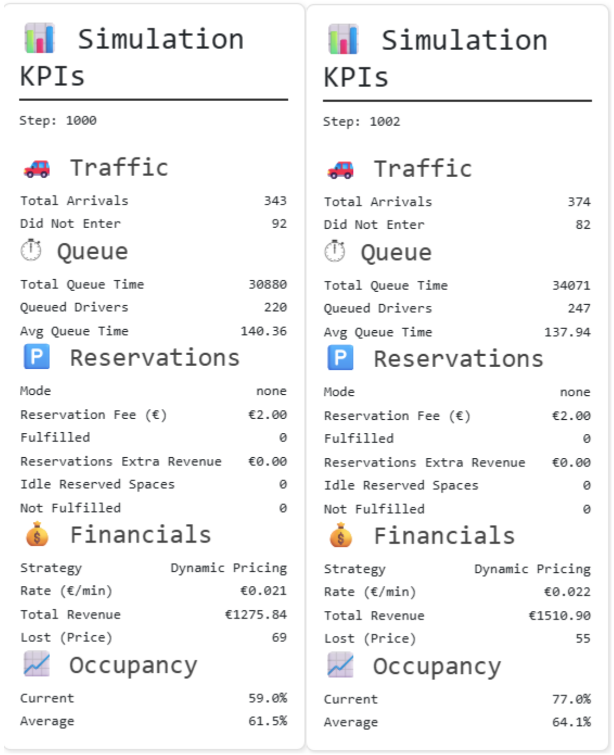
Revenue does not correlate linearly with occupancy. The actual highest occupancy occurs under the dynamic pricing (second run), but the baseline strategy also achieves competitive revenue with much lower congestion therefore maximizing occupancy alone is not an optimal objective. Reservations must be coupled with separate access lanes rules to be effective, which is often difficult to implement in real life scenarios. Demand elasticity is the critical factor for dynamic pricing success. Dynamic pricing and reservation strategies increase waiting times, which leads to the fact that it is important to implement infrastructure changes that allow queue aware admission.



```
📊 Simulation
KPIs
Step: 1000

🚙 Traffic
Total Arrivals                 215
Did Not Enter                   35
⏱ Queue
Total Queue Time              7968
Queued Drivers                 149
Avg Queue Time               53.48
🅿 Reservations
Mode                  reservations
Reservation Fee (€)          €3.00
Fulfilled                       30
Reservations Extra
Revenue                     €90.00
Idle Reserved Spaces             0
Not Fulfilled                    0
💰 Financials
Strategy              Reservations
Rate (€/min)                €0.022
Total Revenue             €1503.72
Lost (Price)                     7
📈 Occupancy
Current                      47.0%
Average                      61.3%
```

Fig. 3. Reservation with 1 lane.



```
📊 Simulation                          📊 Simulation
KPIs                                   KPIs
Step: 1000                             Step: 1002

🚙 Traffic                             🚙 Traffic
Total Arrivals                 343     Total Arrivals                 374
Did Not Enter                   92     Did Not Enter                   82
⏱ Queue                                ⏱ Queue
Total Queue Time             30880     Total Queue Time             34071
Queued Drivers                 220     Queued Drivers                 247
Avg Queue Time              140.36     Avg Queue Time              137.94
🅿 Reservations                        🅿 Reservations
Mode                          none     Mode                          none
Reservation Fee (€)          €2.00     Reservation Fee (€)          €2.00
Fulfilled                        0     Fulfilled                        0
Reservations Extra Revenue   €0.00     Reservations Extra Revenue   €0.00
Idle Reserved Spaces             0     Idle Reserved Spaces             0
Not Fulfilled                    0     Not Fulfilled                    0
💰 Financials                          💰 Financials
Strategy           Dynamic Pricing     Strategy           Dynamic Pricing
Rate (€/min)                €0.021     Rate (€/min)                €0.022
Total Revenue             €1275.84     Total Revenue             €1510.90
Lost (Price)                    69     Lost (Price)                    55
📈 Occupancy                           📈 Occupancy
Current                      59.0%     Current                      77.0%
Average                      61.5%     Average                      64.1%
```

Fig. 4. Dynamic Pricing Metrics, both for the first and second run, respectively.

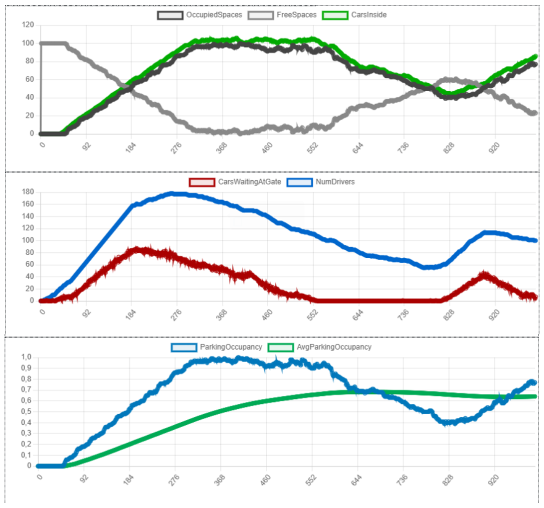


Fig. 5. Dynamic Pricing (Run 2) Strategy Results.

V. Conclusions and Future Work

Static pricing offers the best trade–offs between revenue and congestion, although other strategies can outperform it in pure revenue and occupancy rate, as seen above with the reservation and dynamic pricing strategies. All strategies are constrained by a single entry bottleneck, limiting policy-only solutions.

Regarding the future work, it should be explored the evaluation of other infrastructure layouts, such as multiple gates or priority access, extending demand modeling with learning or memory effects (users reacting to past experiences) and calibrating willingness to pay and arrival patters using any available real world data. On top of that and like it was mentioned before, with more time and effort, the field of implementing deep learning strategies or similar could and will give us more variance, such as giving each driver even more personal attributes, and the results will be even more realistic, further improving the efficacy of these strategies and a better evaluation, therefore a more risk-free implementation.

References

[1] D. Shoup, *The High Cost of Free Parking*. Chicago, IL: Planners Press, American Planning Association, 2005.

[2] T. Rye *et al.*, “16 Good Reasons for Parking Management,” PUSH & PULL Project, supported by Intelligent Energy Europe, Tech. Rep., 2016. [Online].

[3] Python Software Foundation, “Python Language Reference,” [Online]. Available: http://www.python.org.

[4] U. Wilensky, “NetLogo,” Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, 1999. [Online]. Available: http://ccl.northwestern.edu/netlogo/.

[5] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, “SUMO – Simulation of Urban MObility: An Overview,” in *Proc. SIMUL 2011, Third Int. Conf. Adv. Syst. Simul.*, Barcelona, Spain, 2011, pp. 55–60.

[6] Team SimPy, “SimPy: Discrete event simulation for Python,” 2020. [Online]. Available: https://simpy.readthedocs.io.

[7] J. Masad and J. Kazil, “Mesa: Agent-based modeling in Python 3+,” in *Proc. 14th Python in Science Conf. (SCIPY 2015)*, Austin, TX, 2015, pp. 53–60.