

Final Project Report

Wanqi Yuan (wy21) & Jonathan Gan (wg18)

Tue., Apr. 30

Introduction

Problem Statement

The parking garage problem is currently structured to assume that demand is deterministic and constant. This is not representative of real world conditions as demand is highly dynamic and will change in response to various city conditions. It does not consider that the price of a parking spot very much affects the demand for parking. That is, the higher the price of a parking spot, the lower the demand for the spots in that garage. The parking garage problem also currently will “decide” to add another level if demand exceeds capacity at all, but that could be at a margin of even one spot, which is not realistic nor profitable.

For our final project, we want to better capture the variance in demand for parking, particularly in San Francisco. We will explore through research the relationship between price of parking and the demand of parking to implement into a sequential decision problem of whether to build additional levels of a parking garage at a yearly timestep.

Selected Feature

The feature we will include is the relationship between price and demand and how a difference in price will affect total revenue. At each yearly timestep, we will find the optimal amount of levels for the parking garage to exist at to maximize revenue, operating under the assumption that we are always building a capacity to match the demand. This resolves our problem statement by providing a representation that demand is not deterministic and that price is constant regardless of demand.

Demand is a quantifiable variable for many cases outside of parking that determine whether a given action will be taken. By more accurately modeling this relationship between how the price of *something* might change the demand of it, and therefore the revenue and whether the construction that might take place as a result, we can apply this to other climate scenarios

that would require sequential decision making. For example, the decision to expand a solar farm on an annual basis depends on the demand of the energy for the clients it serves and the price of the energy that can be supplied from the solar farm.

Literature Review

Much of existing literature explores the demand side of the problem, using pricing to control demand instead of increasing capacity since varying the capacity-side is often seen as impractical and cost-prohibitive. For instance, Simicevic et al. (2012) utilized a binary logit model to identify the main parameters that influenced user behavior in choosing where to park in Belgrade. They found that in the case of parking garages, as parking cost increases, demand for parking for the average driver would correspondingly decrease. This study proved sensitivity of pricing towards driver behavior, which directly correlates with garage occupancy.

While we recognize that expanding the capacity of parking garages is not always practical, we wanted to incorporate this as an alternative since we wanted to account for all the decisions of the garage owner. For many, the main objective is to maximize profits, and in dense urban areas with high demand, the best way to maximize revenue may be to increase capacity. By incorporating capacity expansion, we aim to create a more comprehensive model for this sequential decision process, evaluating the tradeoffs between initial capital investment and long term revenue gain from increased parking capacity.

Nijsten (2017) explored the optimization of parking capacities in urban areas with the following objectives - minimizing emissions, travel times, and maximizing efficiency in land utilization. Various evolutionary algorithms such as mutation and survivor selection were utilized to solve the optimization problem, applying to small networks. It was found that a rank-based selection method performed the best in optimizing parking capacities for a case study conducted in Delft. In the determination of the initial capacity for construction within our model, we based it on a similar ranking selection method to better model the optimization process.

Methodology

Implementation

The data used to create our demand curve is sourced from the San Francisco Municipal Transit Authority. We utilized entry and exit data at the Union Square parking garage on March 29 2013 to determine demand for parking in relation to the price, as well as garage occupancy. We are assuming that conditions on this day are representative of daily conditions on any given day in the year.

In order to build our price vs. demand curve, many assumptions were made. We took the number of entries/exits to be the total daily demand. We then aimed to calculate a weighted average price for this total demand. We did this by categorizing every entry/exit data point into hourly price buckets varying from \$1-\$7. For example, if one of our entries' price was between \$3.5-\$4.49, it would be categorized as a \$4 cost. We calculated the average price to be the sum of each bucket price times the number of data points within that bucket, divided by the sum of all the data points, 657. This resulting price was at an hourly time step, so we multiplied this price by 12 (assuming that very few people would park for more than half a day), and 365, to get it to a yearly time step price. We also assumed based on the data from San Francisco Municipal Transit Authority that at \$30/hour, the demand for parking would be zero. With these assumptions, we derived the following linear relationship: $\text{demand} = -0.046 * \text{price} + 1500$.

From there, we built a function to optimize the number of levels built starting with demand vs. price relationship at a given year. We start with time year 1, with zero levels built. Within each time step, an original price is taken and at that associated demand, the total revenue is calculated. Included in this analysis is the construction costs per space, which is taken into account when calculating revenue. Litman (2023) found average construction costs for basic parking structures for various US cities in 2022. In San Francisco, it would cost \$30,316 to construct one additional space in an existing parking structure. You then consider within this analysis whether adding one more level that includes 150 more spots results in more revenue. If it does, you build that level. You then repeat the consideration for another level until it is no longer more profitable, having found the optimal number of levels.

At the next time step, you start at the resulting number of levels from the previous time step, but the process of deciding whether to build additional levels remains the same. Additionally, at each time step, the y-intercept increases at a certain percentage to represent overall demand increase for parking. We took this value to be randomly generated from a normal distribution that represents the population increase of San Francisco, with a mean of 0.45% and a standard deviation of 0.1%. This represents the uncertainty of population growth. The slope of the curve is also changing to represent inflation at a certain rate, say, 4%. That is, the slope represents how many spots of demand you lose, at each dollar increases in price. At each time step, we divided the slope by 1.04, to show how with inflation, the number of spots of demand you lose with a dollar increase is less, because the dollar is worth less. We continue this analysis for some number of years, but this can be done for a shorter or longer period with a simple change.

Validation

In order to validate our code, we ran the simulation function with varying values for deterministic parameters including: timestep, growth increment, inflation rate, and construction cost. We did so to qualitatively see if the results made sense with the change in parameters. For example, decreasing the increment (the number of additional spots per level built), should result

in more levels being built initially as well as more frequently as the years progress. The same is true of the opposite. An increased inflation rate would mean more levels being built sooner (since there would be less demand decrease for each dollar decrease), which was reflected in how the code ran. An increase in construction cost per space would also result in less levels being built over time.

Results

With the simulation, we ran the simulation for a total of 9 times, varying each of the 3 main inputs to determine the relationship between each input and the optimal number of levels to build per year.

```
using Plots
using Distributions
Plots.default(; margin=5Plots.mm)
```

```
function get_profit(capacity, m, b, cost)

    # Cost is cost per spot
    price = div(capacity - b, m) # Floor Div for dollars
    return (price * capacity) - (cost * capacity)
end
```

get_profit (generic function with 1 method)

```
function optimize(maxSpots, increment, start_spots, m, cost_per_spot)
    b = maxSpots
    levels_to_add = div(maxSpots - start_spots, increment)
    optLevels = 0
    optProfit = get_profit(start_spots, m, b, cost_per_spot)
    curr_levels = 0
    while curr_levels < levels_to_add
        curr_capacity = start_spots + curr_levels * increment
        profit = get_profit(curr_capacity, m, b, cost_per_spot)
        if profit > optProfit
            optProfit = profit
            optLevels = curr_levels
        end
        curr_levels += 1
    end
end
```

```

    return optLevels
end

```

optimize (generic function with 1 method)

```

function simulation(timesteps, start_spots, increment, demand_growth_rate, init_b, m, inflation_rate)
    y_levels = [] # LEVELS AT TIMETSTEP T
    y_curr_profit = [] # PROFIT AT TIMETSTEP T
    y_net_profit = [] # NET PROFITS
    net_profit = 0
    curr_levels = 0
    timestep = 0
    curr_spots = start_spots
    b = init_b

    while timestep < timesteps
        levels_to_add = optimize(b, increment, curr_spots, m, cost_per_spot)
        if levels_to_add == 0
            # println("Whoops")
        end
        curr_spots += increment * levels_to_add
        curr_levels += levels_to_add

        curr_profit = get_profit(curr_spots, m, b, cost_per_spot) / 1000
        net_profit += curr_profit
        push!(y_curr_profit, curr_profit)
        push!(y_net_profit, net_profit)
        push!(y_levels, curr_levels)

        b *= (1 + demand_growth_rate)
        m /= (1 + inflation_rate)

        timestep += 1
    end

    return y_levels
end

```

simulation (generic function with 1 method)

INCREMENT = 100 Inflation Rate = 0.04 Construction Cost = 30000

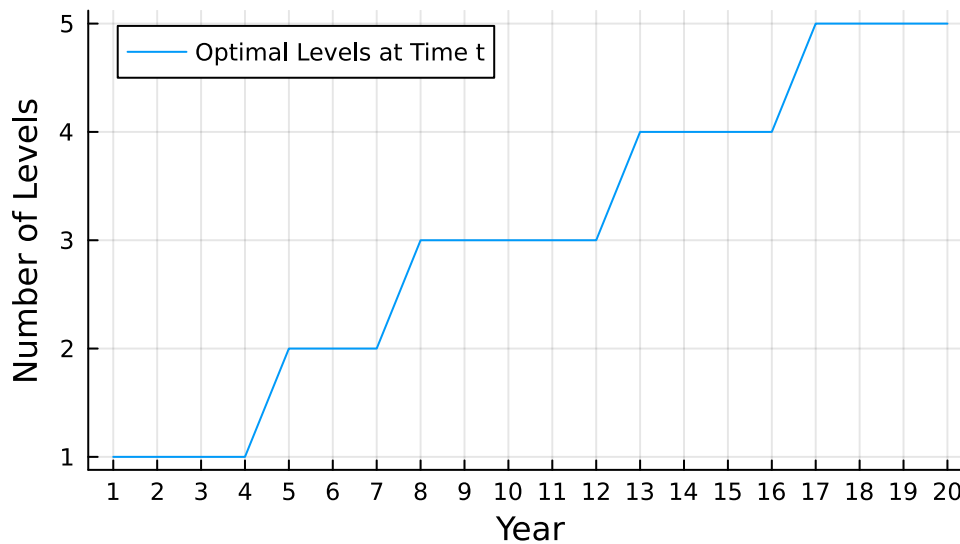
```

timesteps = 20
start_spots = 0
increment = 100
demand_growth_rate = rand(Normal(0.0045, 0.001))
init_b = 1500
m = -0.046
inflation_rate = 0.04
cost_per_spot = 30000
y_levels = simulation(timesteps, start_spots, increment, demand_growth_rate, init_b, m, infla
display_growth_rate = round(demand_growth_rate, sigdigits=3)
plot_title = "Parking Garage Simulation"
font_size = 16
println(y_levels)
plot(y_levels, label="Optimal Levels at Time t", xlabel="Year", ylabel="Number of Levels",
title=plot_title, titlefontsize=font_size, lw=1, gridlinewidth=1, xticks=0:1:25, yticks=0:1:
# plot(y_curr_profit, label="Current Annual Profit at time t")
# plot(y_net_profit, label="Net Profit at time t")
# legend()
#display()

```

Any[1, 1, 1, 1, 2, 2, 2, 3, 3, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5]

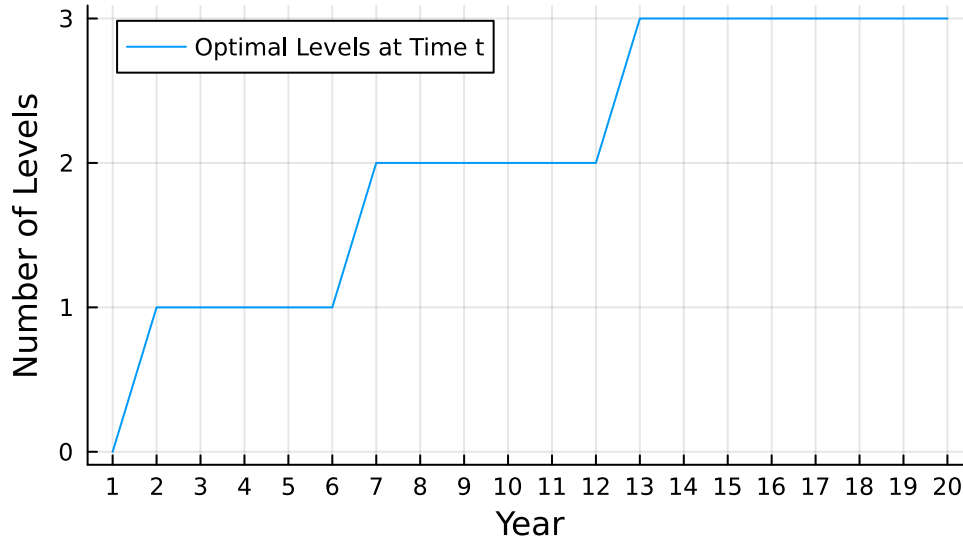
Parking Garage Simulation



INCREMENT = 150 Inflation Rate = 0.04 Construction Cost = 30000

Any[0, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3]

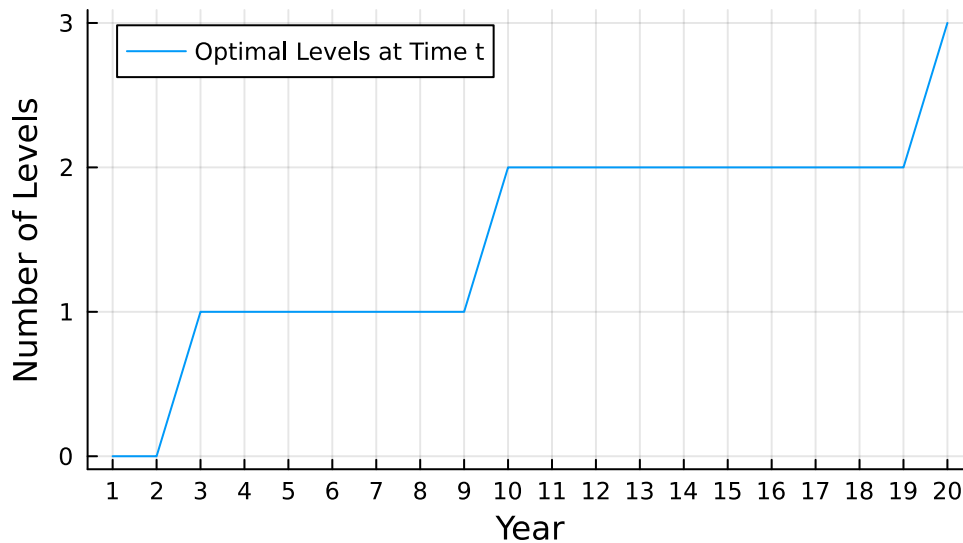
Parking Garage Simulation



INCREMENT = 200 Inflation Rate = 0.04 Construction Cost = 30000

Any[0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3]

Parking Garage Simulation

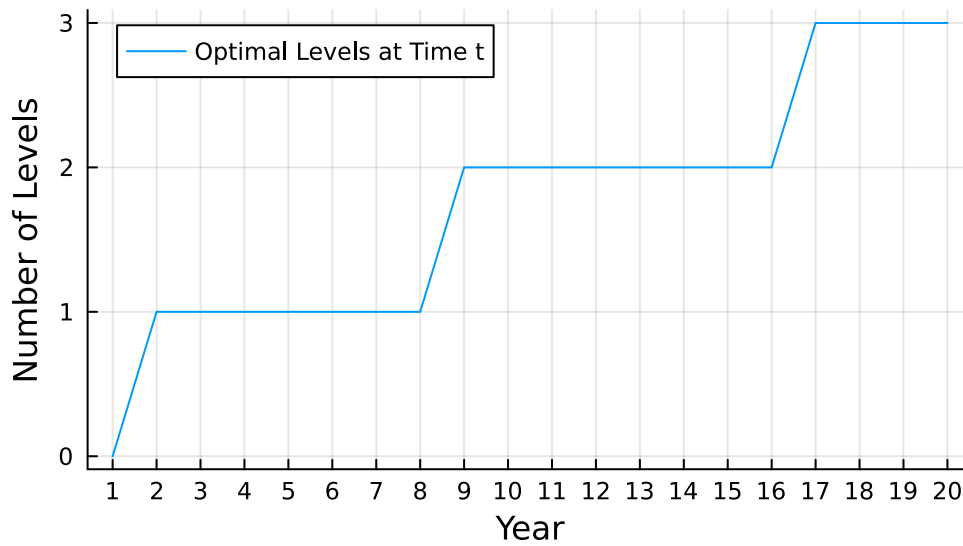


Increment = 150 INFLATION RATE = 0.03 Construction Cost = 30000

```
timesteps = 20
start_spots = 0
increment = 150
demand_growth_rate = rand(Normal(0.0045, 0.001))
init_b = 1500
m = -0.046
inflation_rate = 0.03
cost_per_spot = 30000
y_levels = simulation(timesteps, start_spots, increment, demand_growth_rate, init_b, m, infl
display_growth_rate = round(demand_growth_rate, sigdigits=3)
plot_title = "Parking Garage Simulation"
font_size = 16
println(y_levels)
plot(y_levels, label="Optimal Levels at Time t", xlabel="Year", ylabel="Number of Levels",
title=plot_title, titlefontsize=font_size, lw=1, gridlinewidth=1, xticks=0:1:25, yticks=0:1:
# plot(y_curr_profit, label="Current Annual Profit at time t")
# plot(y_net_profit, label="Net Profit at time t")
# legend()
#display()
```

Any[0, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3]

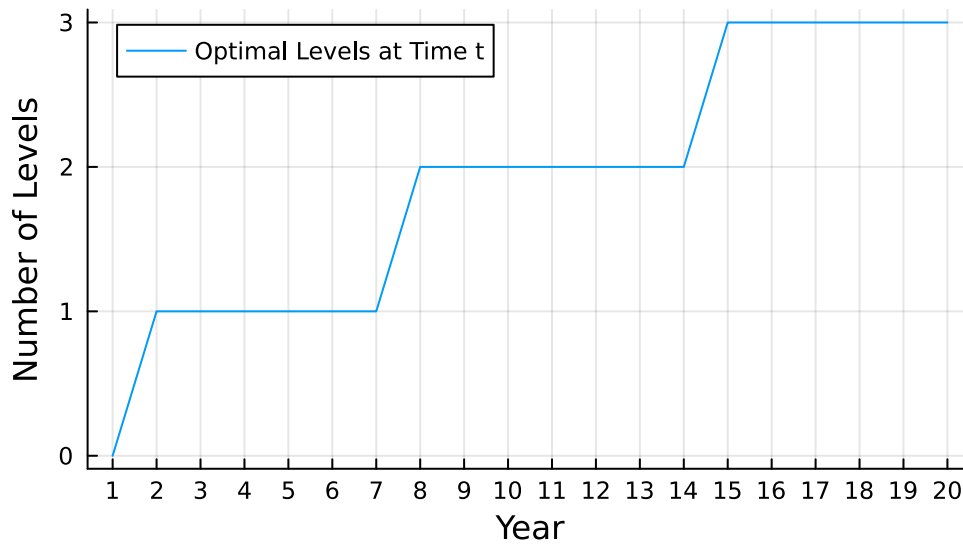
Parking Garage Simulation



Increment = 150 INFLATION RATE = 0.04 Construction Cost = 30000

Any[0, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3]

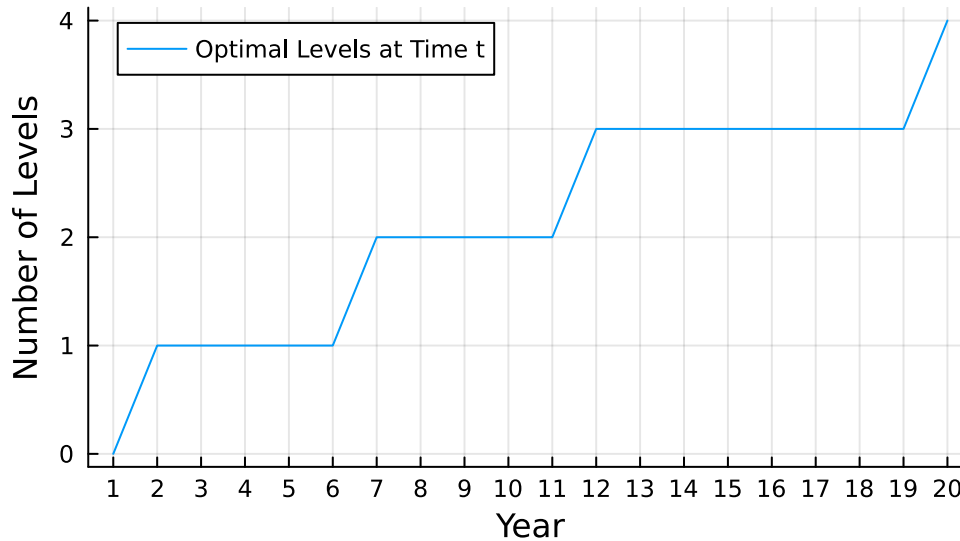
Parking Garage Simulation



Increment = 150 INFLATION RATE = 0.05 Construction Cost = 30000

Any[0, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 4]

Parking Garage Simulation



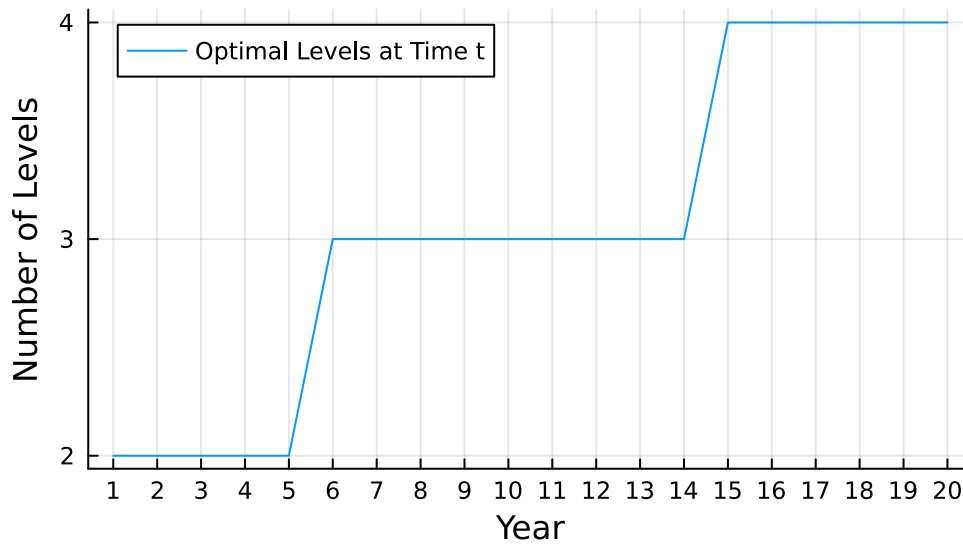
Increment = 150 Inflation Rate = 0.04 CONSTRUCTION COST = 20000

```
timesteps = 20
start_spots = 0
increment = 150
demand_growth_rate = rand(Normal(0.0045, 0.001))
init_b = 1500
m = -0.046
inflation_rate = 0.04
cost_per_spot = 20000
y_levels = simulation(timesteps, start_spots, increment, demand_growth_rate, init_b, m, infl
display_growth_rate = round(demand_growth_rate, sigdigits=3)
plot_title = "Parking Garage Simulation"
font_size = 16
println(y_levels)
plot(y_levels, label="Optimal Levels at Time t", xlabel="Year", ylabel="Number of Levels",
title=plot_title, titlefontsize=font_size, lw=1, gridlinewidth=1, xticks=0:1:25, yticks=0:1:
# plot(y_curr_profit, label="Current Annual Profit at time t")
# plot(y_net_profit, label="Net Profit at time t")
```

```
# legend()
#display()
```

```
Any[2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4]
```

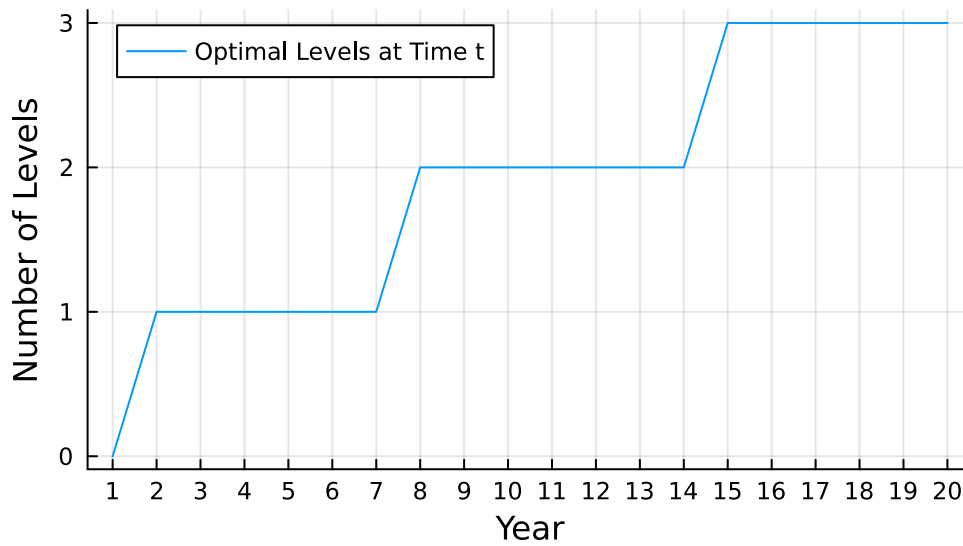
Parking Garage Simulation



Increment = 150 Inflation Rate = 0.04 CONSTRUCTION COST = 30000

```
Any[0, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3]
```

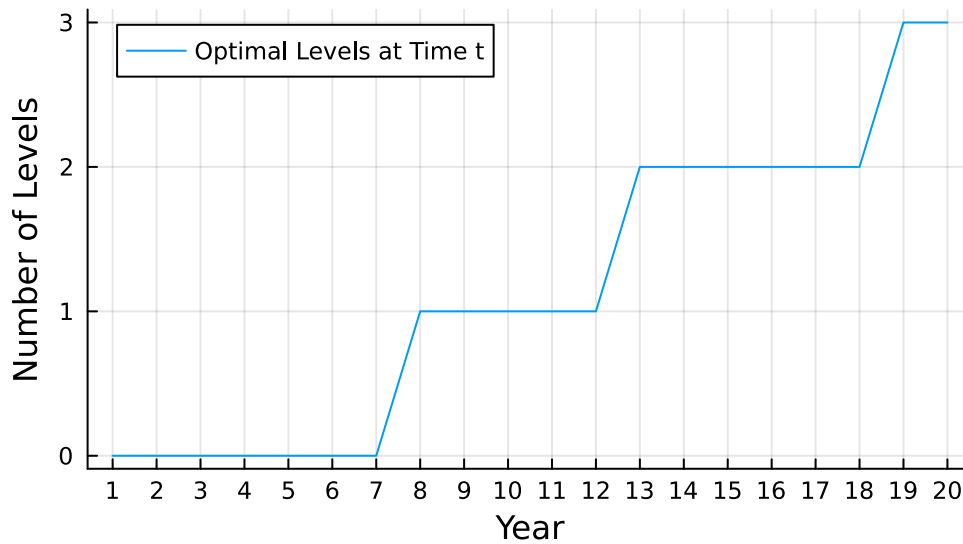
Parking Garage Simulation



Increment = 150 Inflation Rate = 0.04 CONSTRUCTION COST = 40000

Any[0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 3, 3]

Parking Garage Simulation



Conclusions

Discussion

Our results demonstrate the modeling of the relationship between price and demand for a sequential decision problem where demand is a large factor on a given decision. As mentioned briefly in the problem statement, this model structure can be applied to many other cases beyond the parking garage problem. It can be applied to analyses for infrastructure development that supports renewable energy with the goal of building the appropriate size for that infrastructure to support the demands, remaining profitable, but also be resilient against natural disasters. For example, the model could be developed to find the “sweet spot” size for a solar farm where it is large enough to support the demands of the area it is serving, still being profitable, while also being large enough to remain partially operational should a natural disaster result in part of the farming being temporarily under maintenance. It can also provide more guidance on the front end of how changes in population or economical changes can impact how they can anticipate expansion of energy infrastructure. A large limitation to this model is that the way we determine demand is not the most accurate. We assumed that every entry/exit data point was equal to a full 12 hour day of parking in order to build the curve, which is not true as the raw data shows. We also aggregated all data to assume yearly demand and price from one day, which is not accurate or representative. We would suggest a better method to develop a price vs. demand curve.

Conclusions

While the model structure allows for varying parameter changes, if we’re assuming that each level of the parking garage includes 150 spots, a randomly sampled demand increase with a mean of 0.45% and a standard deviation of 0.1%, and an interest rate of 4%, we observed that on average, a new level was built every 4-6 years. It was interesting to see that at the starting demand, it was not profitable to build even the first level until about 4-5 years in. This feature we’ve implemented now demonstrates how the relationship between price and demand will affect the decision at each yearly time step on whether to build an additional level. It also shows how it might take years before it is profitable to build a garage at all with a starting demand and price relationship compared to the costs to build.

In the case of varying parameters, the lower the number of spots per level, the more levels were built sooner since there was a lower threshold of demand needed to justify the additional costs of building a new level of spots. At a higher interest rate, it would also result in building more levels sooner, because the demand decrease per dollar increase, was less and less each year. And with higher construction costs per space, the less levels were built.

Our findings illustrate the correlation between price and demand within a sequential decision-making scenario, where demand significantly influences the decision-making process. This