Disneyland Visit for Maximum Enjoyment Using Linear Programming Model

BIA-650 Final Project

Venkata Krishna Rohan Dacharla (20020527)

Sri Sai Sindhu Penmetsa (20022513)

Alok Kommajesula (20017259)

Sai Sandesh Boorle (20020530)

Problem Overview and Motivation:

Embarking on a Disneyland adventure requires strategic planning to transform the dream into a seamless and joy-filled reality. If done incorrectly, the trip can be filled with long lines and overcrowding causing a nightmare experience. To ensure it's as fun as possible, you have to consider a lot of different factors including the long wait times, crowd levels, and seeing a show at an appropriate time. The goal is to make an itinerary that not only minimizes the wait times and maximizes the fun and enjoyable experience.

Optimization Model

To optimize your visit to Disneyland, our model incorporates the following decision variables:

- Wait times for each attraction.
- Show schedules and attendance.
- Dining reservations.
- Crowd levels in different park areas.

Objective Function:

The primary objective is to maximize overall satisfaction. This is achieved by considering the weighted sum of attraction, show, and dining satisfaction.

Constraints:

- To ensure an optimal experience, constraints are set to meet specified criteria for:
- wait times can't be greater than the time the park is open
- show schedules: there only a few shows therefore we need to be there on time.
- attraction balancing: creating the right priority of attractions that the whole family can enjoy
- crowd levels

Model Assumptions:

To achieve this goal, certain assumptions have been made to streamline decision-making processes. These assumptions, ranging from the linearity of ride waiting times to the predictability of show schedules, are justified for their role in optimizing the overall Disneyland adventure. Moreover, we draw distinctions between our model and those discussed in the course, emphasizing the unique aspects of our approach tailored for the magical realm of Disneyland.

Disneyland Visit for Maximum Enjoyment Using Linear Programming Model

Model Input Parameter Estimation and Data Requirement

- **Attraction Data:**
    - **Source:** The attraction data, including names, start times, end times, ratings, and waiting times, is created within the code using NumPy's random functions.
    - **Parameter Estimation:** The 'Attraction_Rating' and 'Waiting_Time' parameters are generated randomly using **np.random.randint(1, 6, 5)** and **np.random.randint(10, 60, 5)**, respectively.
- **Assigned Parameters:**
    - **Total Available Time (total_available_time):**
        - **Source:** Assigned value based on the problem statement or specific constraints. In this case, it is set to 10 hours.
        - **Assignment:** The parameter is explicitly assigned a value based on the available time for attractions.
    - **Max Total Waiting Time (max_total_waiting_time):**
        - **Source:** Assigned value based on the problem statement or specific constraints. In this case, it is set to 80 minutes.
        - **Assignment:** The parameter is explicitly assigned a value based on the maximum acceptable waiting time.
    - **Weights (df['Weight']):**
        - **Source:** Assigned values based on some criteria or preferences. In this case, they are manually provided as weights for the linear programming model.
        - **Assignment:** The weights are explicitly assigned based on the importance or preference for each attraction.
- **Linear Programming Model Parameters:**
    - **Objective Coefficients (c):**
        - **Source:** Derived from the assigned weights (**df['Weight']**).
        - **Parameter Estimation:** The weights are assigned based on some criteria or preferences, and they influence the optimization model.
    - **Equality Constraint Coefficients (A_eq and b_eq):**
        - **Source:** Assigned based on the problem statement or constraints. In this case, the equality constraint ensures that the total time spent on attractions is equal to the total available time.
        - **Assignment:** Explicitly assigned based on the constraint that the sum of time spent should be equal to the total available time.
    - **Inequality Constraint Coefficients (A_ub and b_ub):**
        - **Source:** Assigned based on the problem statement or constraints. In this case, the inequality constraint ensures that the total waiting time does not exceed the maximum allowed waiting time.
        - **Assignment:** Explicitly assigned based on the constraint that the sum of waiting times should be less than or equal to the maximum total waiting time.
- **Optimization Result:**
    - The **linprog** function is used to solve the linear programming problem and obtain the optimized schedule (**optimized_schedule**) and total enjoyment (**-result.fun**).

Disneyland Visit for Maximum Enjoyment Using Linear Programming Model

In summary, the code combines assigned values, manually assigned weights, and randomly generated attraction data to formulate and solve a linear programming model for optimizing the schedule of visiting attractions. The parameters are determined based on the problem requirements and preferences, and the linear programming model is constructed to achieve the desired optimization outcome.

The Optimization Solution Method

In the world of math and problem-solving, the linear function plays a big part, kind of like the superhero of the math world, called the objective function. Cuemath, an online math guru, says, "Real-life relationships can be super complicated. But guess what? Linear programming can help unravel the mess, making it way easier to figure things out" ("Linear Programming"). For our solution, we used python to help with the model. The libraries used included pandas, numpy, and linprog.

Optimal Solution Structure and Insights

Figure 1

```
import pandas as pd
import numpy as np
from scipy.optimize import linprog
np.random.seed(42)
data = {

'Attraction': ['Space Mountain', 'Pirates of the Caribbean', 'The Lion King Show', 'Haunted Mansion', 'Splash Mountain'],
'Start_Time': [9, 10, 12, 11, 10],
'End_Time': [17, 18, 20, 16, 19],
'Attraction_Rating': np.random.randint(1, 6, 5),
'Waiting_Time': np.random.randint(10, 60, 5),
'Weight': [0.4, 0.3, 0.2, 0.5, 0.4]
}
df = pd.DataFrame(data)
total_available_time = 10  # Total available time in hours
max_total_waiting_time = 80  # Max total waiting time in minutes


# Linear Programming Model

c = -df['Weight'].values
A_eq = [[1, 1, 1, 1, 1]]
b_eq = [total_available_time]
A_ub = -df['Waiting_Time'].values.reshape(1, -1)
b_ub = [-max_total_waiting_time]
result = linprog(c, A_eq=A_eq, b_eq=b_eq, A_ub=A_ub, b_ub=b_ub, method='revised simplex')
```

Python code to optimize enjoyment level considering different shows and timings

Disneyland Visit for Maximum Enjoyment Using Linear Programming Model

Figure 2

```
# Sensitivity analysis for objective function coefficients
obj_sensitivity = result.x

# Sensitivity analysis for right-hand side values of constraints
rhs_sensitivity_eq = result.slack
rhs_sensitivity_ub = result.con
optimized_schedule = pd.DataFrame({'Attraction': df['Attraction'], 'Time_Spent': result.x})

print("Optimized Schedule:")

print(optimized_schedule)

print("\nTotal Enjoyment:", -result.fun)

print("\nObjective Function Sensitivity (Solution Values):", obj_sensitivity)

print("\nConstraint Sensitivity (Slack and Surplus):")

print("Equality Constraints:", rhs_sensitivity_eq)

print("Inequality Constraints:", rhs_sensitivity_ub)
```

Python code for sensitivity analysis

Figure 2

```
Optimized Schedule:
                 Attraction  Time_Spent
0            Space Mountain         2.0
1  Pirates of the Caribbean         1.0
2        The Lion King Show         1.0
3           Haunted Mansion         3.0
4           Splash Mountain         3.0

Total Enjoyment: 3.1
```

Python output for optimisation model

```
Objective Function Sensitivity (Total Enjoyment): 3.1

Constraint Sensitivity (Slack and Surplus):
Equality Constraints: [0.]
Inequality Constraints: [0. 10. 20. 30. 40.]
```

Python output for Sensitivity Analysis

Disneyland Visit for Maximum Enjoyment Using Linear Programming Model

Conclusions, Model Limitations and Extensions

The optimization model designed to maximize enjoyment during a Disneyland visit offers a methodical, data-driven approach to curating an optimal itinerary. Through the minimization of wait times, strategic scheduling of shows, and ensuring a well-rounded experience across attractions, the model aims to elevate visitor satisfaction and deliver a seamless and enchanting Disneyland experience.

However, it's essential to acknowledge the limitations of the model. Factors such as unpredictable weather, unexpected ride closures, and varying preferences among visitors may introduce uncertainties that the model cannot fully account for. Additionally, individual preferences for specific attractions or spontaneous changes in plans may challenge the model's precision.

Looking ahead, there is potential for model extension and refinement. Incorporating real-time data, such as live queue updates or personalized visitor profiles, could enhance the model's adaptability and responsiveness to dynamic conditions. Despite its current limitations, the optimization model lays the groundwork for continuous improvement, emphasizing the ongoing pursuit of an even more magical and tailored Disneyland experience.

References

1. Winston, Wayne L. and S. Christian Albright, Practical Management Science, Pacific Grove, CA: Duxbury 2018 (6th edition).
2. Disneyland. "Entertainment." Disneyland Resort, n.d. Web. 15 Dec. 2023. <https://disneyland.disney.go.com/entertainment/#/sort=alpha/>.
3. Stojiljković, Mirko. "Introduction to Linear Programming with Python and PuLP." Real Python, DIOGENeS L, 1 Jan. 2022, https://realpython.com/linear-programming-python/.