Optimizing a Multi-Day Travel Itinerary Under Cost and Scheduling Constraints

Ashlee Liu

May 10, 2025

Contents

1 Introduction

For this project, I designed a 3-day travel itinerary to **Sequoia National Park**, located in California's southern Sierra Nevada mountains. They are renowned for their giant sequoia trees, scenic hiking trails, and natural landmarks. My goal was to plan a cost-efficient, activity-filled road trip that balanced recreation and rest under realistic travel and lodging constraints.

1.1 Data Collection Process

Collected real-world cost and availability data for lodging, car rental, and park activities:

- **Lodging:** Researched lodging options from official *park websites* and *Expedia*. Considered rooms that fit at least two adults, selecting (*Wuksachi Lodge* and *Montecito Sequoia Lodge*). Lodging costs were rounded to the nearest integer, and captured a price range to model uncertainty. Breakfast costs were included as either an add-on or separate food budget based on lodge's offerings.
- **Transportation:** Selected car rental options by filtering for automatic transmission vehicles priced less than $100, using *Expedia* listings for pickup and drop-off at San Francisco on May 26 and 28. Considered three cars (Chevrolet Beat, Chevrolet Aveo, Volkswagen Vento) with similar fuel efficiency to focus the optimization on rental cost.
- **Activities:** Selected activity options from official park websites and *AllTrails*, prioritizing diverse range of hiking trails and tours. Each activity was characterized by type, distance, duration, intensity, suggester (hypothetically, two people planning), time of day, and additional fees. The Congress Trail was modeled as an extension of the Sherman Tree Trail based on park layout, imposing a dependency constraint, and the timing for some activities imposed a constraint on which days we could assign to.

1.2 Modeling Assumptions

Simplifying assumptions to enable feasible and realistic optimization model:

- **Lodging prices are uncertain** within published ranges, used maximum estimated price to model a worst-case scenario.
- **Car rental fuel costs were excluded** because the vehicles' fuel efficiency ratings were similar and driving distances within the park are relatively short.
- **Activity times were categorized** as sunrise, afternoon, or night to reflect practical scheduling limits (e.g., stargazing only occurs at night).
- **Time windows for each day were specified** based on standard check-in (4 PM) and check-out (11 AM) times at the lodges, as well as realistic daylight hours for park activities.

Day	Time Window	Total Hours
Day 1	5 PM – 10 PM	5 hours
Day 2	7 AM – 8 PM (2 hour break)	11 hours
Day 3	8 AM – 5 PM (2 hour break)	7 hours

Table 1: Daily Available Time Windows and Total Activity Hours

2 Modeling

2.1 Sets, Parameters, and Decision Variables

2.1.1 Sets

- L: Lodging selection (*Wuksachi Lodge*, *Montecito Sequoia Lodge*) impacts total cost and models uncertainty through worst-case pricing.
- C: Car choice (Beat, Aveo, or Vento) determines transportation cost.
- A: Set of activities (Sherman Tree Trail, Congress Trail, Stargazing, Watchtower Trail, Marble Falls Trail, Crystal Cave Tour, Moro Rock Trail, Big Trees Trail, Alta Peak Trail) structure trip itinerary and impact daily workload, costs, and enjoyment.
- D: Set of days ($D = \{1, 2, 3\}$) to organize scheduling decisions across three trip days.

2.1.2 Parameters

- lodging_cost_i: Cost per night for lodging option $i \in L$, including taxes and fees.
- breakfast_cost_i: Breakfast cost per day for lodging option $i \in L$.
- car_cost_i: Total rental cost for car option $i \in C$, including taxes and fees for the entire trip.
- activity_cost_i: Additional cost to participate in activity $i \in A$ (e.g., guided tours, stargazing programs).
- duration_i: Duration (in hours) of activity $i \in A$.
- time_of_day_i: Time category (Sunrise, Afternoon, Night) when activity $i \in A$ must be scheduled.
- suggester_i: Identifier indicating which person suggested activity $i \in A$ (1 or 2).
- available_hours_i: Number of hours available on day $i \in D$ for activities, considering breaks and travel constraints.
- max_downtime_i: Maximum allowable downtime (in hours) on day $i \in D$ between activities.
- intensity_i: Physical intensity level for activity $i \in A$ (1 = Easy, 2 = Moderate, 3 = Strenuous).

2.1.3 Decision Variables

- $\text{use\_lodge}_i \in \{0, 1\}$: Binary variable indicating whether lodging option $i \in L$ is selected.
- $\text{use\_car}_i \in \{0, 1\}$: Binary variable indicating whether car rental option $i \in C$ is selected.
- $\text{schedule}_{i,d} \in \{0, 1\}$: Binary variable indicating whether activity $i \in A$ is scheduled on day $d \in D$.
- $\text{day\_hours}_d \geq 0$: Total number of hours of scheduled activities on day $d \in D$.
- $\text{fatigue\_penalty}_d \geq 0$: Fatigue penalty incurred on day $d \in D$ based on deviation from the target activity intensity.
- $\text{intensity\_sum}_d \geq 0$: Sum of the intensity scores of all activities scheduled on day $d \in D$.

2.2 Constraints

1. **Lodging and Car Selection:** Exactly one lodging option $i \in L$ and one car rental option $i \in C$ must be selected:

$$\sum_{i \in L} \text{use\_lodge}_i = 1, \quad \sum_{i \in C} \text{use\_car}_i = 1$$

2. **Total Daily Activity Hours:** For each day $d \in D$, the total scheduled hours must equal the sum of the durations of the scheduled activities:

$$\text{day\_hours}_d = \sum_{i \in A} \text{schedule}_{i,d} \times \text{duration}_i$$

3. **Daily Time Limit:** The total scheduled activity hours must not exceed the available hours on day $d \in D$:

$$\text{day\_hours}_d \leq \text{available\_hours}_d$$

4. **Downtime Constraint:** The scheduled hours must cover at least the available hours minus the allowed maximum downtime:

$$\text{day\_hours}_d \geq \text{available\_hours}_d - \text{max\_downtime}_d$$

5. **Single Assignment of Activities:** Each activity $i \in A$ can be scheduled on at most one day:

$$\sum_{d \in D} \text{schedule}_{i,d} \leq 1$$

6. **No Sunrise Activities on Day 1:** Activities categorized as "Sunrise" cannot be scheduled on Day 1:

$$\text{schedule}_{i,1} = 0 \quad \text{for all } i \text{ with } \text{time\_of\_day}_i = \text{Sunrise}$$

7. **No Night Activities on Day 3:** Activities categorized as "Night" cannot be scheduled on Day 3:

$$\text{schedule}_{i,3} = 0 \quad \text{for all } i \text{ with } \text{time\_of\_day}_i = \text{Night}$$

8. **Inclusion of Other Person's Suggested Activity:** Each day must include at least one activity suggested by the other person:

$$\sum_{i \in A : \text{suggester}_i = 2} \text{schedule}_{i,d} \geq 1$$

9. **Congress Trail Requires Sherman Tree Trail:** If Congress Trail is scheduled, Sherman Tree Trail must also be scheduled on the same day:

$$\text{schedule}_{\text{Congress},d} \leq \text{schedule}_{\text{Sherman},d}$$

10. **No Two Strenuous Hikes on Consecutive Days:** No two strenuous activities (intensity 3) may be scheduled on consecutive days:

$$\sum_{i \in A : \text{intensity}_i = 3} \text{schedule}_{i,d} + \sum_{i \in A : \text{intensity}_i = 3} \text{schedule}_{i,d+1} \leq 1 \quad \text{for } d = 1, 2$$

11. **At Least One Activity Scheduled:** At least one activity must be scheduled across all days:

$$\sum_{i \in A} \sum_{d \in D} \text{schedule}_{i,d} \geq 1$$

2.3 Fatigue Penalty Modeling

To model traveler fatigue, I introduced an auxiliary variable fatigue_penalty_d based on the deviation of daily activity intensity from a target value of 5.

The total scheduled intensity for each day is computed as:

$$\text{intensity\_sum}_d = \sum_{i \in A} \text{schedule}_{i,d} \times \text{intensity}_i$$

The fatigue penalty for each day is defined as the absolute deviation from the target intensity:

$$\text{fatigue\_penalty}_d \geq \text{intensity\_sum}_d - 5$$

$$\text{fatigue\_penalty}_d \geq 5 - \text{intensity\_sum}_d$$

Linearization: As absolute value functions are nonlinear, the penalty is linearized using two separate inequalities to ensure compatibility with the linear programming solver. With the penalty, the model discourages schedules that are either too easy or too strenuous relative to the ideal activity level.

2.4 Objective Function

Objective of optimization model is to minimize total trip cost, combining monetary expenses and fatigue penalties. The objective function is:

$$\min \left(2 \sum_{i \in L} \text{use\_lodge}_i \times \text{lodging\_cost}_i + \sum_{i \in C} \text{use\_car}_i \times \text{car\_cost}_i + \sum_{i \in A} \sum_{d \in D} \text{schedule}_{i,d} \times \text{activity\_cost}_i + 10 \sum_{d \in D} \text{fatigue\_penalty}_d \right)$$

This objective function includes:

- **Lodging Costs:** Multiplied by 2 nights, using the maximum estimated cost to account for price uncertainty.
- **Car Rental Costs:** Fixed total rental cost over the trip duration.
- **Activity Costs:** Additional fees for selected activities such as tours or programs.
- **Fatigue Penalty:** Penalty of 10 units per hour based on deviation from the target daily intensitys.

The minimized objective value reflects the combined cost of lodging, car rental, activity participation, and fatigue impacts under worst-case assumptions. Optimizing this value results in a trip that balances lodging and transportation costs, selects a diverse, affordable set of activities, and encourages daily itineraries that avoid exhaustion.

3 Results: Trip Plan

3.1 Optimized Trip Schedule

Time	Activity
4:00 PM – 5:00 PM	Check-in at Wuksachi Lodge
5:00 PM – 7:00 PM	Crystal Cave Tour (Afternoon, 2.0 hrs)
7:00 PM – 8:00 PM	Big Trees Trail (Afternoon, 1.0 hr)
8:00 PM – 9:00 PM	Free Time / Dinner
9:00 PM – 10:00 PM	Wonders of the Night Sky (Stargazing) (Night, 1.0 hr)

Table 2: Day 1: Arrival and Evening Activities

Time	Activity
7:00 AM – 11:30 AM	Watchtower Trail (Sunrise, 4.5 hrs)
11:30 AM – 3:00 PM	Marble Falls Trail (Sunrise, 3.5 hrs)
3:00 PM – 5:00 PM	Free Time / Rest / Lunch Break (2.0 hrs)
5:00 PM – 6:30 PM	Sherman Tree Trail (Afternoon, 1.5 hrs)
6:30 PM – 7:30 PM	Congress Trail (Afternoon, 1.0 hr)

Table 3: Day 2: Full Hiking Day

Time	Activity
8:00 AM	Check-out from Wuksachi Lodge
8:15 AM – 3:15 PM	Alta Peak Trail (Morning and Afternoon, 7.0 hrs)
3:15 PM – 5:00 PM	Free Time / Rest / Lunch Break

Table 4: Day 3: Morning Hike and Departure

- **Selected Lodge:** Wuksachi Lodge
- **Selected Car Rental:** Chevrolet Beat Sedan
- **Total Objective Value (Cost + Penalties):** $911.00

4 Discussion and Recommendations

Overall, I am satisfied with the solution produced by the optimization model. The final itinerary incorporates a variety of park activities across different difficulty levels and times of day, while maximizing the available hours each day within realistic scheduling constraints. The model effectively balances cost, enjoyment, and physical fatigue.

The optimization approach aligns closely with how I plan trips manually. I typically compare lodging and transportation options by price and amenities, then I generate a list of activities I would be interested in. However, this model differs from my manual planning in that it systematically schedules activities across all available time windows rather than relying on flexible, spontaneous decisions. I normally prioritize a couple "must-do" activities and leave time open to fill with activities based on how I'm feeling. The model provides a more structured itinerary, eliminating the need for impromptu adjustments.

The model could be improved by incorporating additional budgeting elements such as estimating lunch and dinner costs, selecting meal options to minimize expenses. I could also optimize the drive to Sequoia National Park, factoring in potential rest breaks and finding the cheapest gas stations along the way. This would create a more comprehensive optimization model, covering both in-park and en-route logistics.

5 Appendix

5.1 Data Collected

Lodging Name	Cost ($)	Room Type	Breakfast Cost	Amenities
Wuksachi Lodge	313 - 352	2 queens (sleeps 5)	$21.99/person	Buffet restaurant
Montecito Sequoia Lodge	279 - 479	1 queen + bunk	$60 total (2 mornings)	On-site boating, tennis, yoga

Table 5: Lodging Options

Car Model	Total Cost ($)	Seating Capacity	City MPG	Highway MPG
Chevrolet Beat Sedan	87	5	32	40
Chevrolet Aveo	90	5	32	40
Volkswagen Vento	93	5	31	39

Table 6: Car Rental Options

Activity Name	Type	Distance (mi)	Duration (hrs)	Intensity	Proposer	Time	Cost ($)
Sherman Tree Trail	Hike	–	1.5	1	Person 1	Afternoon	0
Congress Trail	Hike	2.0	1.0	2	Person 2	Afternoon	0
Wonders of Night Sky	Stargazing	–	1.0	1	Person 2	Night	20
Watchtower Trail	Hike	8.3	4.5	2	Person 1	Sunrise	0
Marble Falls Trail	Hike	6.3	3.5	2	Person 2	Sunrise	0
Crystal Cave Tour	Tour	0.5	2.0	3	Person 2	Afternoon	20
Moro Rock	Hike	0.6	0.75	3	Person 2	Sunrise	0
Big Trees Trail	Hike	1.0	1.0	1	Person 1	Afternoon	0
Alta Peak Trail	Hike	14.9	7.0	3	Person 2	Afternoon	0

Table 7: Activity List

5.2 Pyomo Formulation

```
if 'google.colab' in sys.modules:
    %pip install pyomo >/dev/null 2>/dev/null
    %pip install highspy >/dev/null 2>/dev/null

solver = 'appsi_highs'

import pyomo.environ as pyo
SOLVER = pyo.SolverFactory(solver)

assert SOLVER.available(), f"Solver {solver} is not available."

model = pyo.ConcreteModel()

# sets
model.L = pyo.Set(initialize=['Wuksachi', 'Montecito'])
model.C = pyo.Set(initialize=['Beat', 'Aveo', 'Vento'])
model.A = pyo.Set(initialize=[
    'Sherman', 'Congress', 'Stargazing', 'Watchtower', 'MarbleFalls',
    'CrystalCave', 'MoroRock', 'BigTrees', 'AltaPeak'
])
model.D = pyo.Set(initialize=[1, 2, 3]) # Days

# params
lodging_cost = {'Wuksachi': 352, 'Montecito': 479}
breakfast_cost = {'Wuksachi': 40, 'Montecito': 60}
```

```
car_cost = {'Beat': 87, 'Aveo': 90, 'Vento': 93}
activity_duration = {
    'Sherman': 1.5, 'Congress': 1.0, 'Stargazing': 1.0, 'Watchtower': 4.5,
    'MarbleFalls': 3.5, 'CrystalCave': 2.0, 'MoroRock': 0.75,
    'BigTrees': 1.0, 'AltaPeak': 7.0
}
activity_cost = {
    'Sherman': 0, 'Congress': 0, 'Stargazing': 40, 'Watchtower': 0,
    'MarbleFalls': 0, 'CrystalCave': 40, 'MoroRock': 0,
    'BigTrees': 0, 'AltaPeak': 0
}
activity_time = {
    'Sherman': 'Afternoon', 'Congress': 'Afternoon', 'Stargazing': 'Night', 'Watchtower': 'Sunrise',
    'MarbleFalls': 'Sunrise', 'CrystalCave': 'Afternoon', 'MoroRock': 'Sunrise',
    'BigTrees': 'Afternoon', 'AltaPeak': 'Afternoon'
}
suggested_by = {
    'Sherman': 1, 'Congress': 2, 'Stargazing': 2, 'Watchtower': 1,
    'MarbleFalls': 2, 'CrystalCave': 2, 'MoroRock': 2,
    'BigTrees': 1, 'AltaPeak': 2
}
available_hours = {1: 5, 2: 11, 3: 7}
max_downtime = {1: 1, 2: 1, 3: 1}

activity_intensity = {
    'Sherman': 1,  # Easy
    'Congress': 2, # Moderate
    'Stargazing': 1, # Easy
    'Watchtower': 2, # Moderate
    'MarbleFalls': 2,# Moderate
    'CrystalCave': 3,# Strenuous
    'MoroRock': 3, # Strenuous
    'BigTrees': 1, # Easy
    'AltaPeak': 3  # Strenuous
}

# pyo params
model.lodging_cost = pyo.Param(model.L, initialize=lodging_cost)
model.breakfast_cost = pyo.Param(model.L, initialize=breakfast_cost)
model.car_cost = pyo.Param(model.C, initialize=car_cost)
model.activity_cost = pyo.Param(model.A, initialize=activity_cost)
model.duration = pyo.Param(model.A, initialize=activity_duration)
model.time_of_day = pyo.Param(model.A, initialize=activity_time, within=pyo.Any)
model.suggester = pyo.Param(model.A, initialize=suggested_by)
model.available_hours = pyo.Param(model.D, initialize=available_hours)
model.max_downtime = pyo.Param(model.D, initialize=max_downtime)
model.intensity = pyo.Param(model.A, initialize=activity_intensity)

# decision var
model.use_lodge = pyo.Var(model.L, domain=pyo.Binary)
model.use_car = pyo.Var(model.C, domain=pyo.Binary)
model.schedule = pyo.Var(model.A, model.D, domain=pyo.Binary)
model.day_hours = pyo.Var(model.D, domain=pyo.NonNegativeReals)
model.fatigue_penalty = pyo.Var(model.D, domain=pyo.NonNegativeReals)
model.intensity_sum = pyo.Var(model.D, domain=pyo.NonNegativeReals)

#constraints
# 1 one lodge and one car selected
model.one_lodge = pyo.Constraint(expr=sum(model.use_lodge[l] for l in model.L) == 1)
model.one_car = pyo.Constraint(expr=sum(model.use_car[c] for c in model.C) == 1)

# 2 Total activity hours per day
def calc_day_hours(m, d):
    return m.day_hours[d] == sum(m.schedule[a, d] * m.duration[a] for a in m.A)
```

```
model.calc_hours = pyo.Constraint(model.D, rule=calc_day_hours)

# 3 Daily time limit
model.daily_limit = pyo.ConstraintList()
for d in model.D:
    model.daily_limit.add(model.day_hours[d] <= model.available_hours[d])

# 4 Downtime constraint
model.downtime_limit = pyo.ConstraintList()
for d in model.D:
    model.downtime_limit.add(model.day_hours[d] >= model.available_hours[d] - model.max_downtime[d])

# 5 Each activity scheduled on at most one day
def one_day_per_activity(m, a):
    return sum(m.schedule[a, d] for d in m.D) <= 1
model.once_per_activity = pyo.Constraint(model.A, rule=one_day_per_activity)

# 6 No sunrise hikes on Day 1
model.sunrise_day1_forbidden = pyo.ConstraintList()
for a in model.A:
    if activity_time[a] == 'Sunrise':
        model.sunrise_day1_forbidden.add(model.schedule[a, 1] == 0)

# 7 No night activities on Day 3
model.night_restriction = pyo.Constraint(expr=model.schedule['Stargazing', 3] == 0)

# 8 Each day must include 1 activity suggested by other person
model.other_person_daily = pyo.ConstraintList()
for d in model.D:
    model.other_person_daily.add(sum(model.schedule[a, d] for a in model.A if suggested_by[a] == 2) >= 1)

# 9 Congress Trail implies Sherman Trail on same day
model.congress_requires_sherman = pyo.ConstraintList()
for d in model.D:
    model.congress_requires_sherman.add(model.schedule['Congress', d] <= model.schedule['Sherman', d])

# 10 No two strenuous hikes on consecutive days
model.no_consecutive_strenuous = pyo.ConstraintList()

for i in range(1, 3):
    model.no_consecutive_strenuous.add(
        sum(model.schedule[a, i] for a in model.A if model.intensity[a] == 3) +
        sum(model.schedule[a, i+1] for a in model.A if model.intensity[a] == 3)
        <= 1
    )

# 11 At least one activity must be scheduled
model.at_least_one_activity = pyo.Constraint(
    expr=sum(model.schedule[a, d] for a in model.A for d in model.D) >= 1
)

# Fatigue penalty is absolute deviation from target intensity 5

def calc_intensity_sum(m, d):
    return m.intensity_sum[d] == sum(m.schedule[a, d] * m.intensity[a] for a in m.A)
model.calc_intensity_sum = pyo.Constraint(model.D, rule=calc_intensity_sum)

def fatigue_abs_rule_1(m, d):
    return m.fatigue_penalty[d] >= model.intensity_sum[d] - 5

def fatigue_abs_rule_2(m, d):
    return m.fatigue_penalty[d] >= 5 - model.intensity_sum[d]

model.fatigue_abs_1 = pyo.Constraint(model.D, rule=fatigue_abs_rule_1)
```

```
model.fatigue_abs_2 = pyo.Constraint(model.D, rule=fatigue_abs_rule_2)

#obj func
model.obj = pyo.Objective(
    expr=(
        sum(model.use_lodge[l] * model.lodging_cost[l] * 2 for l in model.L) +
        sum(model.use_car[c] * model.car_cost[c] for c in model.C) +
        sum(model.schedule[a, d] * model.activity_cost[a] for a in model.A for d in model.D) +
        10 * sum(model.fatigue_penalty[d] for d in model.D) # linear penalty
    ),
    sense=pyo.minimize
)

# solve
results = SOLVER.solve(model)

print("Selected Lodge:")
for l in model.L:
    if pyo.value(model.use_lodge[l]) > 0.5:
        print(f" - {l}")

print("Selected Car:")
for c in model.C:
    if pyo.value(model.use_car[c]) > 0.5:
        print(f" - {c}")

# ordering by
time_priority = {'Sunrise': 0, 'Morning': 1, 'Afternoon': 2, 'Night': 3}

print("Activity Schedule:")
for d in model.D:
    print(f"\nDay {d}:")

    sorted_activities = sorted(
        [a for a in model.A if pyo.value(model.schedule[a, d]) > 0.5],
        key=lambda a: time_priority[model.time_of_day[a]]
    )

    if not sorted_activities:
        print(" (No activities scheduled)")
    else:
        for a in sorted_activities:
            duration = pyo.value(model.duration[a])
            print(f" - {a} ({model.time_of_day[a]}, {duration} hrs)")

print("Daily Hours:")
for d in model.D:
    print(f" Day {d}: {pyo.value(model.day_hours[d]):.2f} hrs")


print(f"Total Objective Value (Lodge + Bfast + Car Rental + Activities + Fatigue Penalty):
    ${pyo.value(model.obj):,.2f}")
```

5.3 Pyomo Output

```
Selected Lodge:
 - Wuksachi
Selected Car:
 - Beat
Activity Schedule:

Day 1:
  - CrystalCave (Afternoon, 2.0 hrs)
  - BigTrees (Afternoon, 1.0 hrs)
  - Stargazing (Night, 1.0 hrs)

Day 2:
  - Watchtower (Sunrise, 4.5 hrs)
  - MarbleFalls (Sunrise, 3.5 hrs)
  - Sherman (Afternoon, 1.5 hrs)
  - Congress (Afternoon, 1.0 hrs)

Day 3:
  - AltaPeak (Afternoon, 7.0 hrs)
Daily Hours:
 Day 1: 4.00 hrs
 Day 2: 10.50 hrs
 Day 3: 7.00 hrs
Total Objective Value (Lodge + Bfast + Car Rental + Activities + Fatigue Penalty): $911.00
```