Optimization problem: Santa’s Workshop Tour

Lana Caldarevic

17th December 2020

https://www.kaggle.com/c/santa-workshop-tour-2019

Agenda

- Problem statement
- Insights into data
- Approach 1
- Approach 2
- Q&A

Problem statement

- optimally schedule visitors for Santa’s Workshop for 100 days before Christmas
- 5000 families have each submitted 10 date preferences
- number of visitors each day between 125 and 300
- If we assign a family to their less desired preference day we have to offer larger and larger gifts as penalty -> <u>preference cost</u>
- <u>Accounting cost</u>

Costs

- Additional cost to Santa (preference cost):
  - choice_0: *no consolation gifts*
  - choice_1: one $50 gift card to Santa's Gift Shop
  - choice_2: one $50 gift card, and 25% off Santa's Buffet (value $9) for each family member
  - choice_3: one $100 gift card, and 25% off Santa's Buffet (value $9) for each family member
  - choice_4: one $200 gift card, and 25% off Santa's Buffet (value $9) for each family member
  - choice_5: one $200 gift card, and 50% off Santa's Buffet (value $18) for each family member
  - choice_6: one $300 gift card, and 50% off Santa's Buffet (value $18) for each family member
  - choice_7: one $300 gift card, and free Santa's Buffet (value $36) for each family member
  - choice_8: one $400 gift card, and free Santa's Buffet (value $36) for each family member
  - choice_9: one $500 gift card, and free Santa's Buffet (value $36) for each family member, and 50% off North Pole Helicopter Ride tickets (value $199) for each family member
  - otherwise: one $500 gift card, and free Santa's Buffet (value $36) for each family member, and free North Pole Helicopter Ride tickets (value $398) for each family member

Costs

- Additional cost to Santa (accounting cost):

$$\sum_{d=100}^{1} \frac{(N_d - 125)}{400} N_d^{\left(\frac{1}{2} + \frac{|N_d - N_{d+1}|}{50}\right)}$$

*Santa's accountants have also developed an empirical equation for cost to Santa that arise from many different effects such as reduced shopping in the Gift Shop when it gets too crowded, extra cleaning costs, a very complicated North Pole tax code, etc.

- N_d is the occupancy of the current day, and N_{d+1} is the occupancy of the *previous* day (we're counting backwards from Christmas!)
- $N_{101}=N_{100}$

Final cost = preference cost + accounting cost

Insights

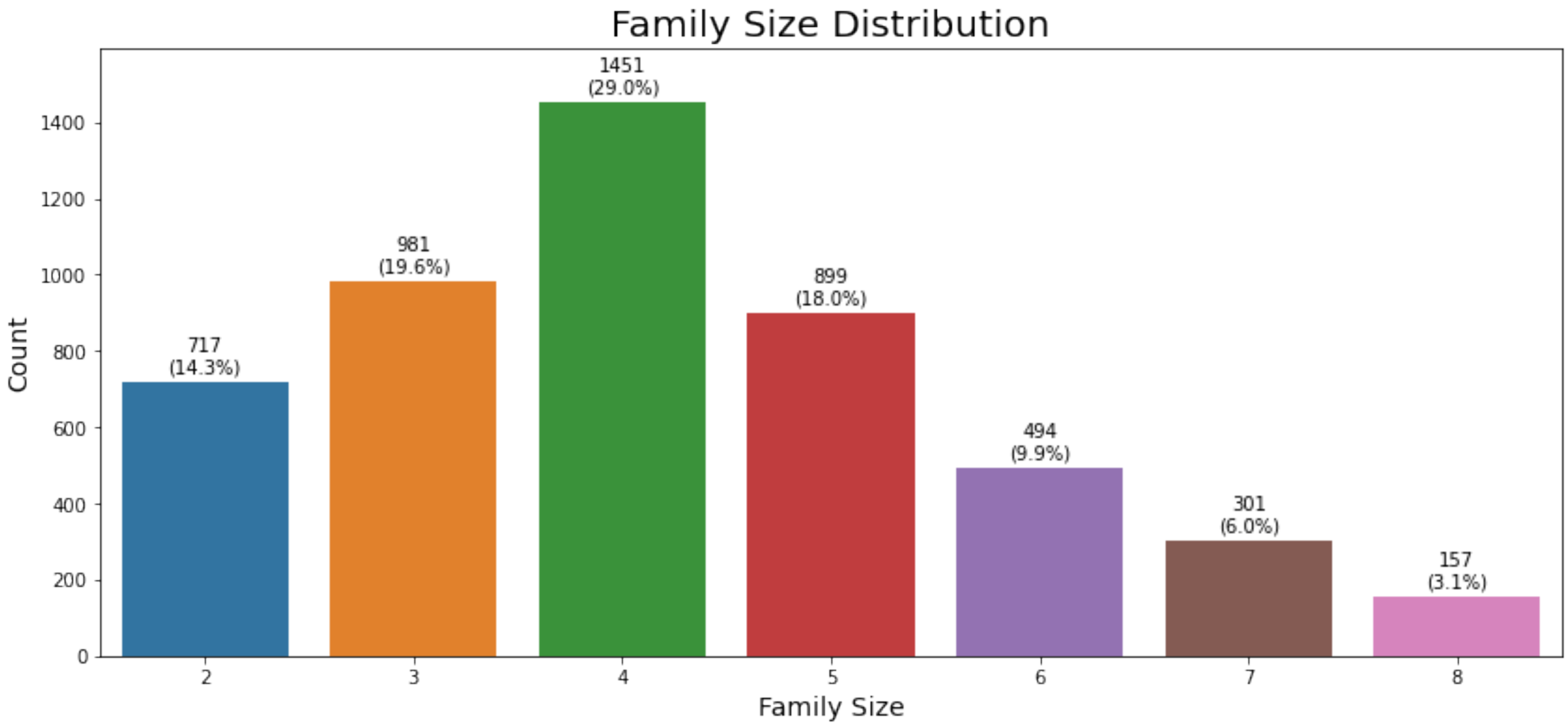
Data

	family_id	choice_0	choice_1	choice_2	choice_3	choice_4	choice_5	choice_6	choice_7	choice_8	choice_9	n_people
0	0	52	38	12	82	33	75	64	76	10	28	4
1	1	26	4	82	5	11	47	38	6	66	61	4
2	2	100	54	25	12	27	82	10	89	80	33	3
3	3	2	95	1	96	32	6	40	31	9	59	2
4	4	53	1	47	93	26	3	46	16	42	39	4
...	...	...	...	...	...	...	...	...	...	...	...	...
4995	4995	16	1	66	33	18	70	56	46	86	60	4
4996	4996	88	66	20	17	26	54	81	91	59	48	2
4997	4997	32	66	54	17	27	21	74	81	3	7	6
4998	4998	67	92	4	17	53	77	1	12	26	70	5
4999	4999	13	11	25	80	88	40	96	39	18	47	4

5000 rows × 12 columns

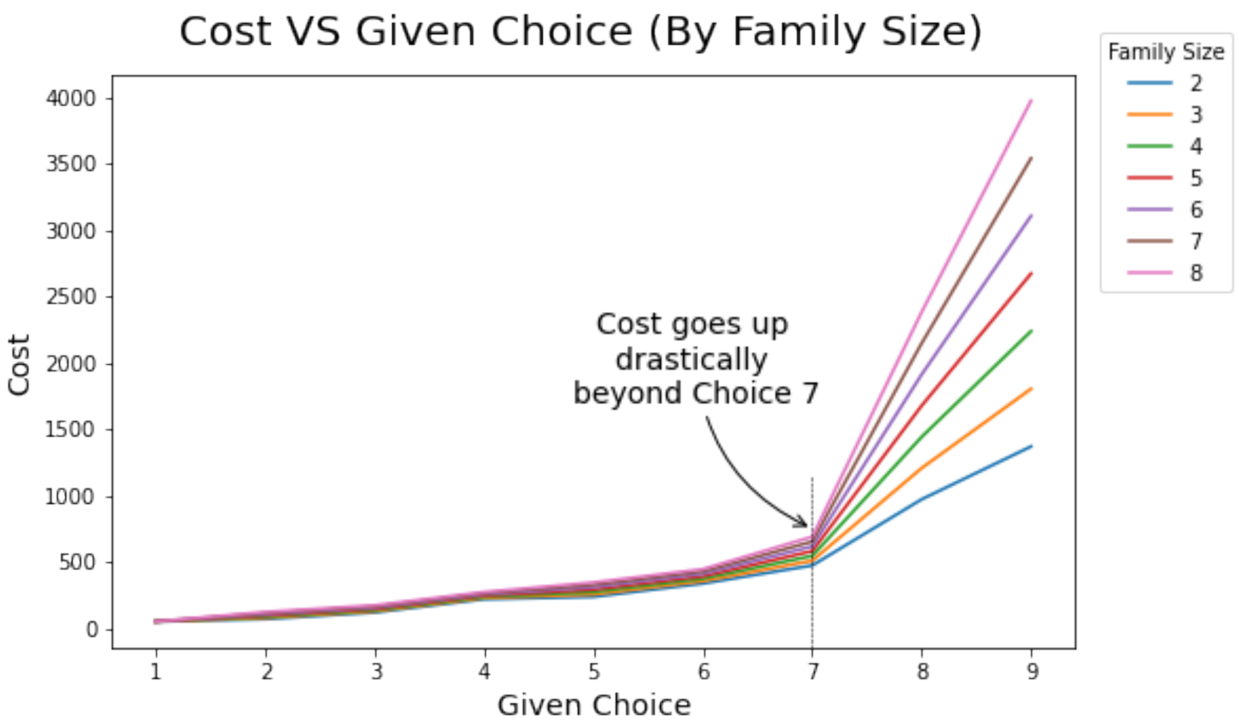
Insights

Family distribution



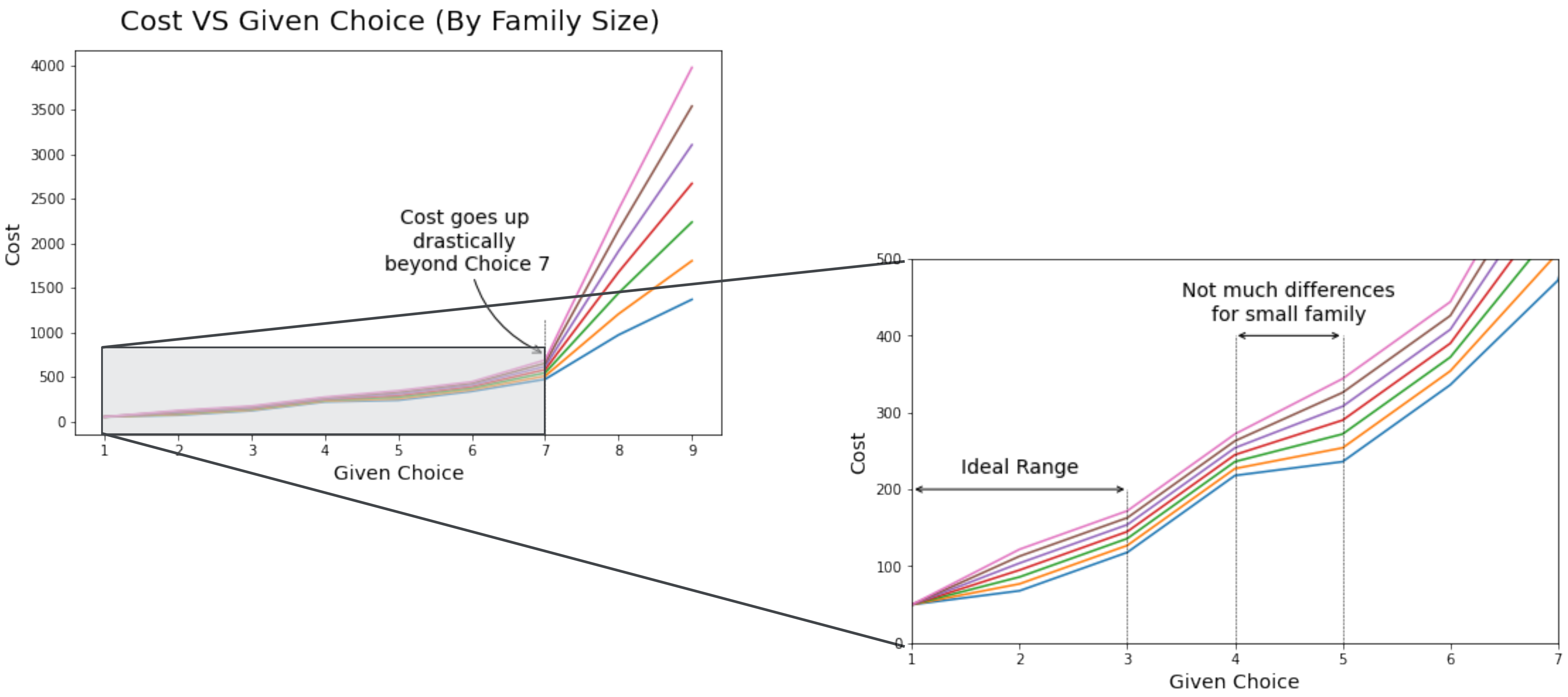
Insights

Preference costs given family size



Insights

Preference costs given family size

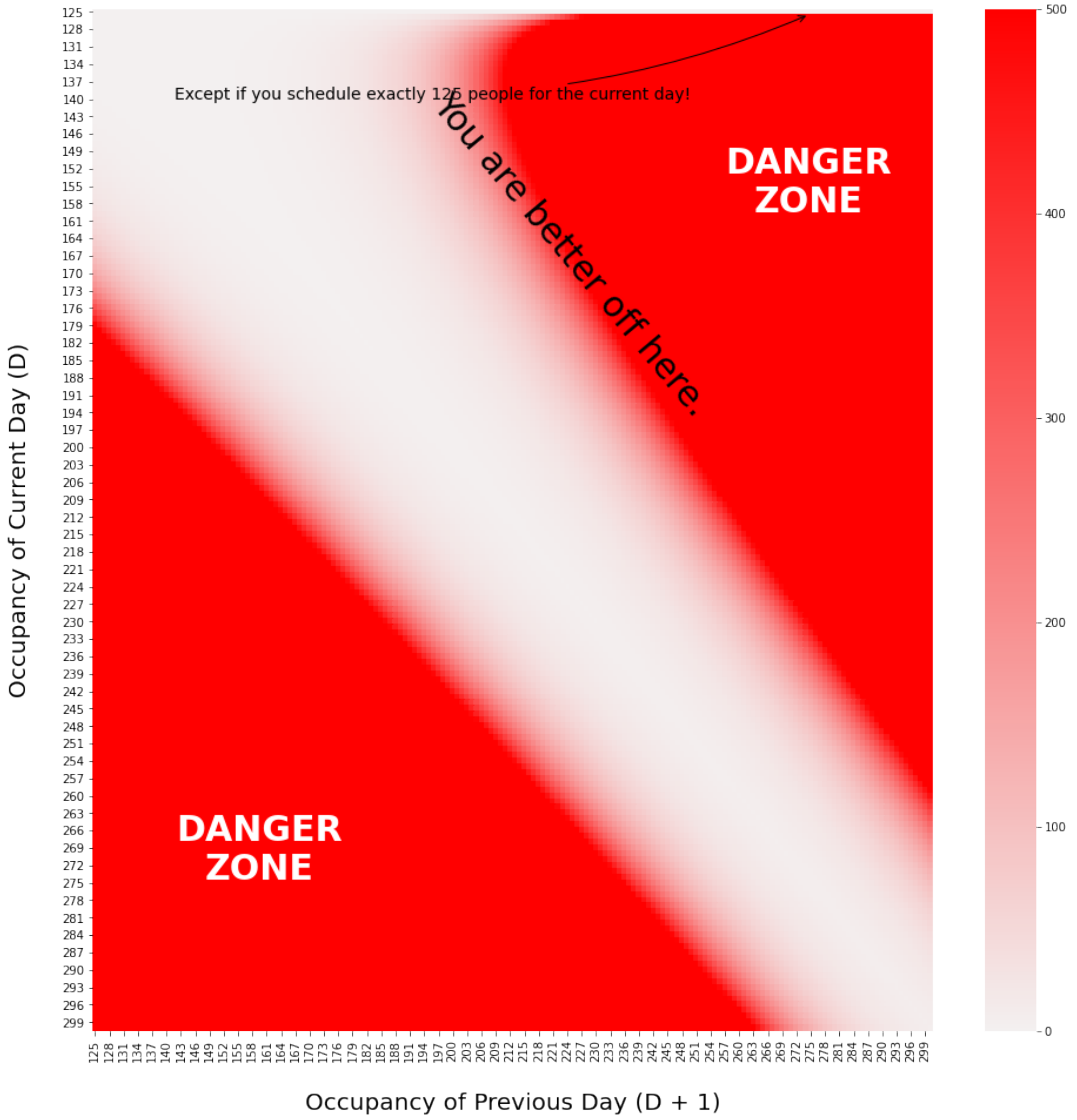


Insights

Accounting cost - details

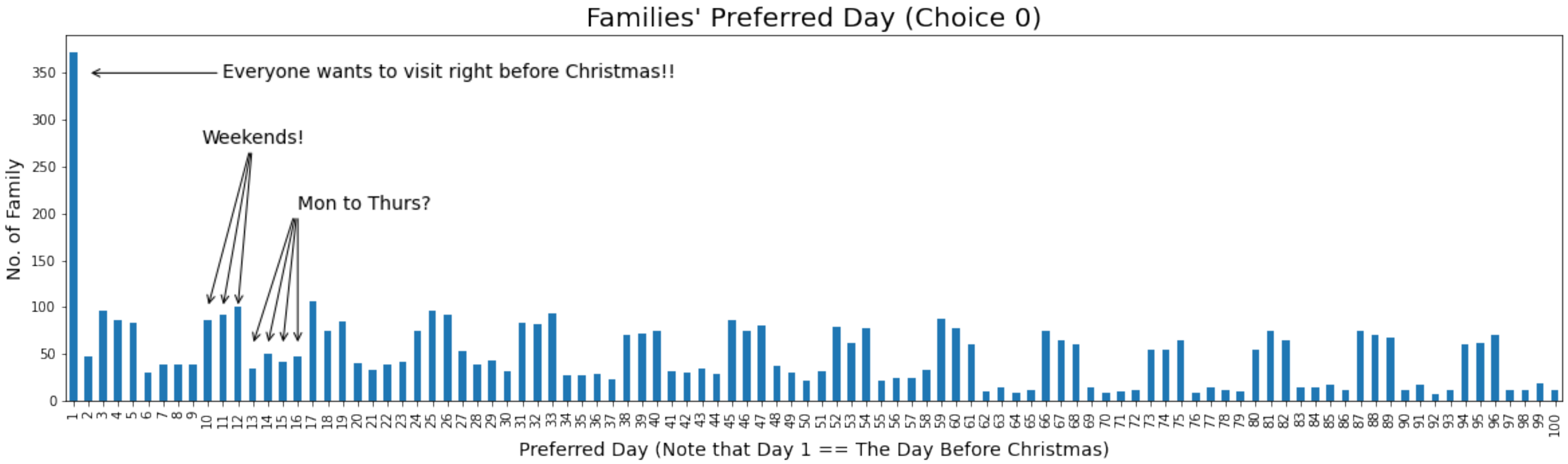
- Make sure the number of people attending everyday is well-balanced
- And if you are unable to do so, make sure that the increase or decrease happens GRADUALLY over the days
- NEVER goes from full house today to the minimum capacity the next day, AVOID the DANGER ZONE unless... You can get exactly 125 people to attend the next day

$$\sum_{d=100}^{1} \frac{(N_d - 125)}{400} N_d^{\left(\frac{1}{2}+\frac{|N_d - N_{d+1}|}{50}\right)}$$



Insights

First choice



Solution

Attempt 1: ignore accounting cost -> optimize only preference cost

- accounting cost = non-linear and non-convex-> challenge for optimization models, let's ignore it for now!
- D - days, F – families
- x_{df}=1, when family *f* is assigned to day *d*, and x_{df}=0 otherwise
- s_f - size of family *f*
- c_{df} - penalty cost incurred when assigning family *f* to day *d* based on their preference list
- Mixed integer programming task
- Max 5000x10 variables, but can be reduced (check cost given family size slide)

$$\text{minimize} \sum_{d\in\mathcal{D}}\sum_{f\in\mathcal{F}} x_{df}\cdot c_{df}$$

subject to:

$$\sum_{f\in\mathcal{F}} s_f x_{df} \geq 125 \quad ,\forall d\in\mathcal{D}$$

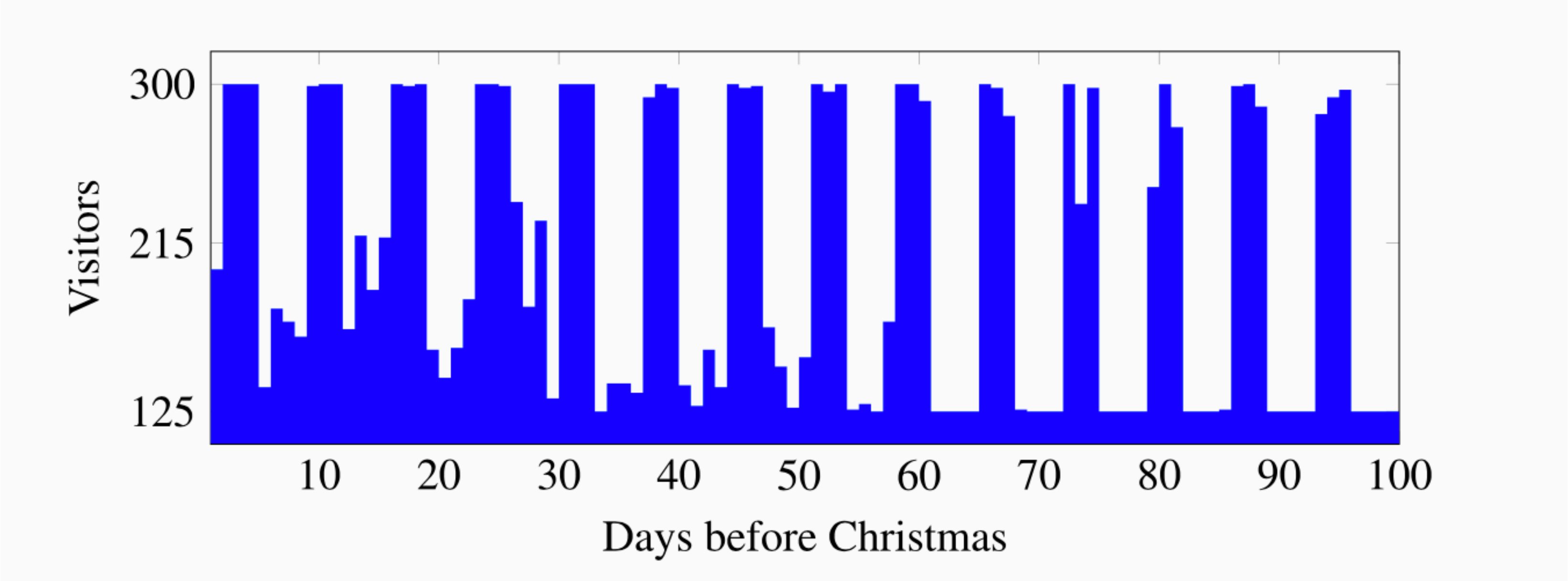
$$\sum_{f\in\mathcal{F}} s_f x_{df} \leq 300 \quad ,\forall d\in\mathcal{D}$$

$$\sum_{d\in\mathcal{D}} x_{df} = 1 \quad ,\forall f\in\mathcal{F}$$

Solution

Attempt 1: ignore accounting cost -> optimize only preference cost

- Cost: (including accounting cost – not optimized)
  13_403_337_617



- Python library: PULP
- Solver: PULP_CBC_CMD (Coin-or)
- Branch and cut algorithm

Solution

Attempt 2: including accounting cost

- Linearizing accounting cost = linear programming problem
- Based on states we could be in for a single day (=the visitor levels) we can transition to a new state for the next day
- State = number of visitors for one day
- Accounting cost = transition cost from one state to other
- -> Graphical representation

$$\sum_{d=100}^{1} \frac{(N_d - 125)}{400} N_d^{\left(\frac{1}{2} + \frac{|N_d - N_{d+1}|}{50}\right)}$$

Solution

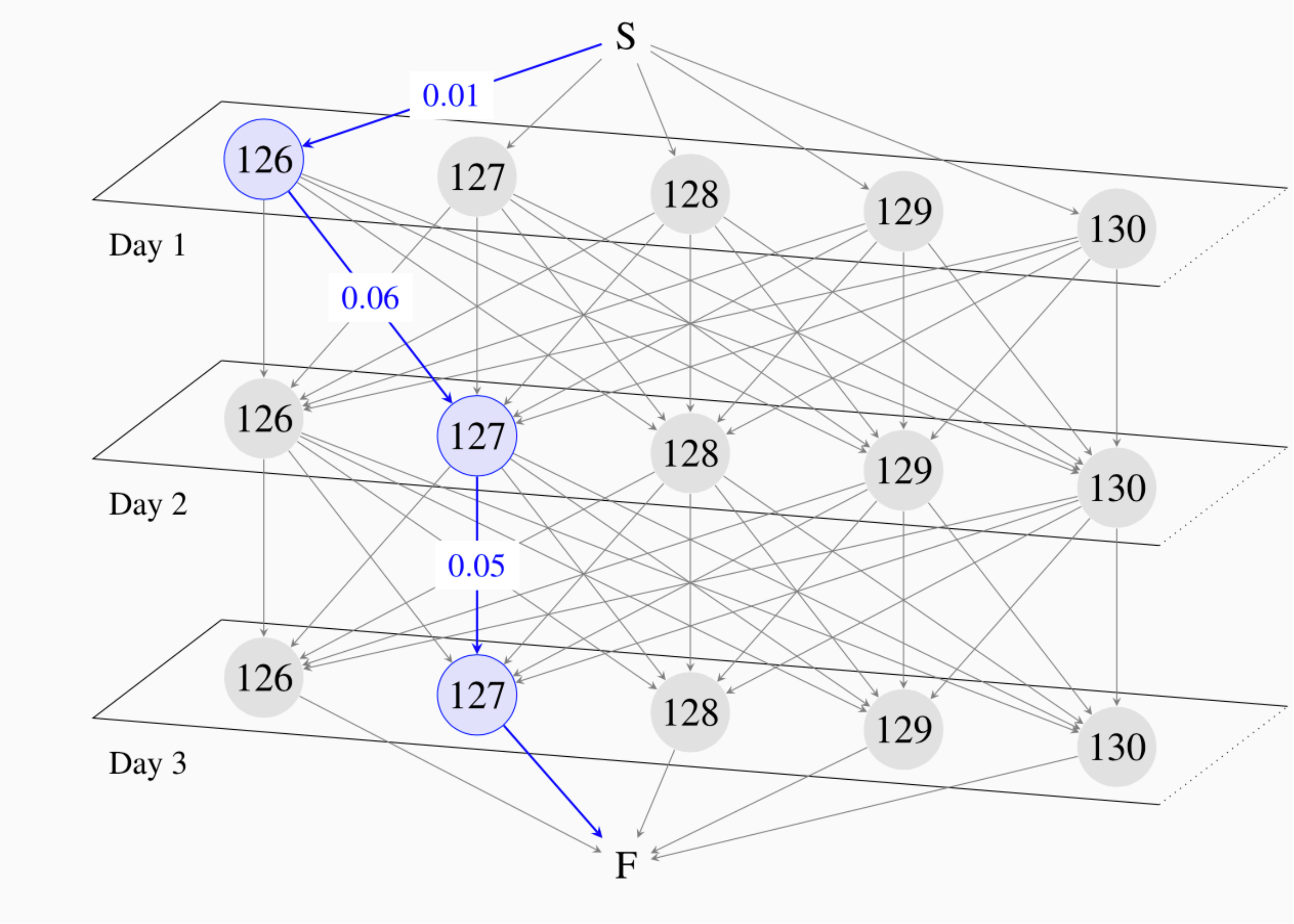
Attempt 2: including accounting cost

$$\sum_{d=100}^{1} \frac{(N_d - 125)}{400} N_d^{\left(\frac{1}{2}+\frac{|N_d - N_{d+1}|}{50}\right)}$$

- G=(V,A) directed graph
- V = {(d,l) | d∈D, l∈L}
- A= {((d,l),(d+1,m))| d∈D, l∈L, m∈L}
- For each arc (i,j) we define a cost:

$$\bar{c}_{ij} = \frac{(j-125)}{400} j^{\left(\frac{1}{2}+\frac{|j-i|}{50}\right)}$$

- (300−125)∗100=175_000 vertices
- ~ 3 mil arcs
- S – source, F –sink node
- -> Minimum cost flow problem: shortest path from the source to the sink node using the costs c_{ij} as distance measure



Solution

Attempt 2: linearizing the accounting cost

- Min cost flow:
- Binary variable a_{ij} indicating if we traversed arc (i,j) or not
- Sets δ_i^+, δ_i^- contain incoming and outgoing arcs of an vertex i respectively

$$
\begin{aligned}
\sum_{j\in\delta_i^-} a_{ij} - \sum_{j\in\delta_i^+} a_{ji} &= 0 \quad &&, \forall i \in V, i \notin \{S, F\} \\
\sum_{j\in\delta_i^-} a_{ij} &= 1 \quad &&, i = S \\
\sum_{j\in\delta_i^+} a_{ji} &= 1 \quad &&, i = F
\end{aligned}
$$

- **Connection** to family assignment problem:

$$
\begin{aligned}
&\textbf{minimize} \sum_{d\in\mathcal{D}} \sum_{f\in\mathcal{F}} x_{df} \cdot c_{df} \\
&\textbf{subject to:} \\
&\sum_{f\in\mathcal{F}} s_f x_{df} \geq 125 \quad , \forall d \in \mathcal{D} \\
&\sum_{f\in\mathcal{F}} s_f x_{df} \leq 300 \quad , \forall d \in \mathcal{D} \\
&\sum_{d\in\mathcal{D}} x_{df} = 1 \quad , \forall f \in \mathcal{F}
\end{aligned}
$$

- linking constraints: assign l visitors on day d when there is an incoming flow to vertex (d,l)

$$
\sum_{f\in\mathcal{F}} s_f x_{df} = \sum_{l\in\mathcal{L}} \sum_{j\in\delta^+_{(d,l)}} l \cdot a_{j,(d,l)} \quad , \forall d \in \mathcal{D}
$$

Solution

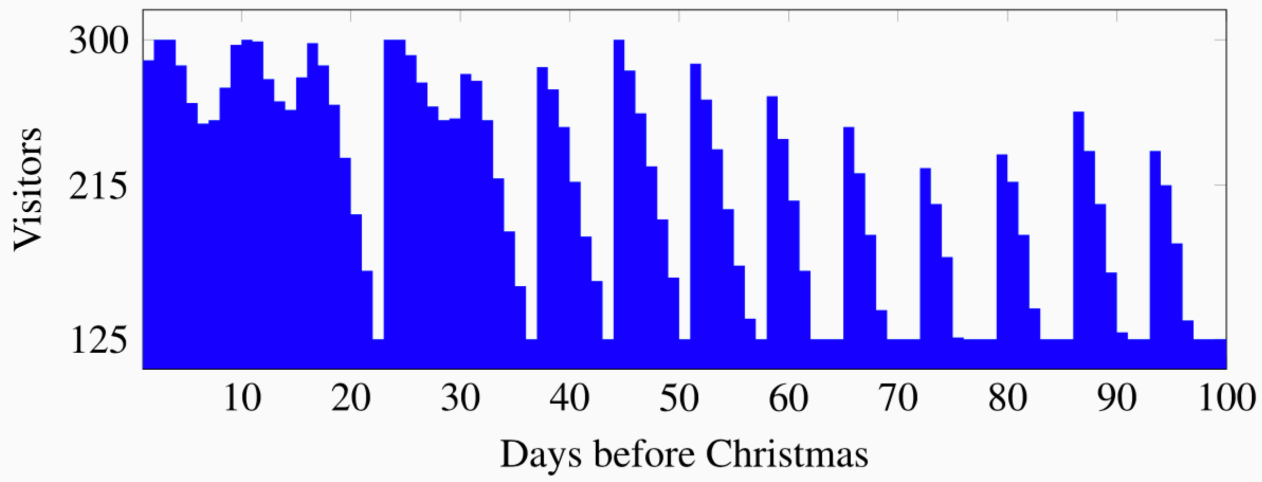
Attempt 2: linearizing the accounting cost

$$
\begin{aligned}
&\text{minimize} \sum_{d\in\mathcal{D}}\sum_{f\in\mathcal{F}} x_{df}\cdot c_{df} \quad + \sum_{(i,j)\in\mathcal{A}} a_{ij}\cdot \bar{c}_{ij} \\
&\text{subject to:} \\
&\sum_{f\in\mathcal{F}} s_f x_{df} && \geq 125 \quad ,\forall d\in\mathcal{D} \\
&\sum_{f\in\mathcal{F}} s_f x_{df} && \leq 300 \quad ,\forall d\in\mathcal{D} \\
&\sum_{d\in\mathcal{D}} x_{df} && = 1 \quad ,\forall f\in\mathcal{F} \\
&\sum_{j\in\delta_i^-} a_{ij} - \sum_{j\in\delta_i^+} a_{ji} && = 0 \quad ,\forall i\in\mathcal{V}, i\notin\{S,F\} \\
&\sum_{j\in\delta_i^-} a_{ij} && = 1 \quad ,i = S \\
&\sum_{j\in\delta_i^+} a_{ji} && = 1 \quad ,i = F \\
&\sum_{f\in\mathcal{F}} s_f x_{df} && = \sum_{l\in\mathcal{L}}\sum_{j\in\delta_i^+} l\cdot a_{j,(d,l)} \quad ,\forall d\in\mathcal{D}
\end{aligned}
$$

Solution

Attempt 2: linearizing the accounting cost

- Solving model: tbd
- Kaggle confirmed optimal cost: **68_888.04343**
- Assignments per day:



Questions?