Work Scheduling for a Retail Chain

Felipe Calderon, Nikki Gong, Rebecca Peng, Vivian Peng

Agenda

1. Background about retail workforce scheduling and relevant researches
2. Problem Statement
3. Conceptual Framework of the Model
4. Mathematical Models
5. Conclusions

Background: Retail Workforce Scheduling

- Brick-and-mortar employed 15% of the American workforce.
- In retail, staff schedules typically change every day and every week with three to seven days of notice of the next week's schedule.
- Most retailers operate under the assumptions that stabilizing employees' schedules would hurt the store's financial performance
- A few leading retailers who adopt a more data-driven approach captured between **4 and 12 % in cost savings** among other facets of store operations.

Relevant Researches on Workforce Scheduling

Lam et al. (1998): Derive a computer-supported scheduling method based on **store traffic forecasting**; our project is based on his proposed model.

Muslius et al. (2002): Use approaches, such as, network flow, the integration of management science and artificial intelligence techniques, and mixed approaches which combine **constraint satisfaction and local improvement algorithm**.

Ernst et al. (2004): Propose a survey of general labor scheduling models which shows the methods of **constraint programming, metaheuristics, and mathematical programming approaches such as set covering/partitioning and AI**

Bard et al. (2003): staff scheduling in the service industry is a multi-faceted assignment problem with many variations that depend on union contracts, company policies, and government regulations. Adding more constraints, such as, ratio constraints **adding reality and complexity** to the formulation.

Problem Statement

Current solutions produce generic schedules fail to account for store-specific factors and workload fluctuations. The undesired results include:

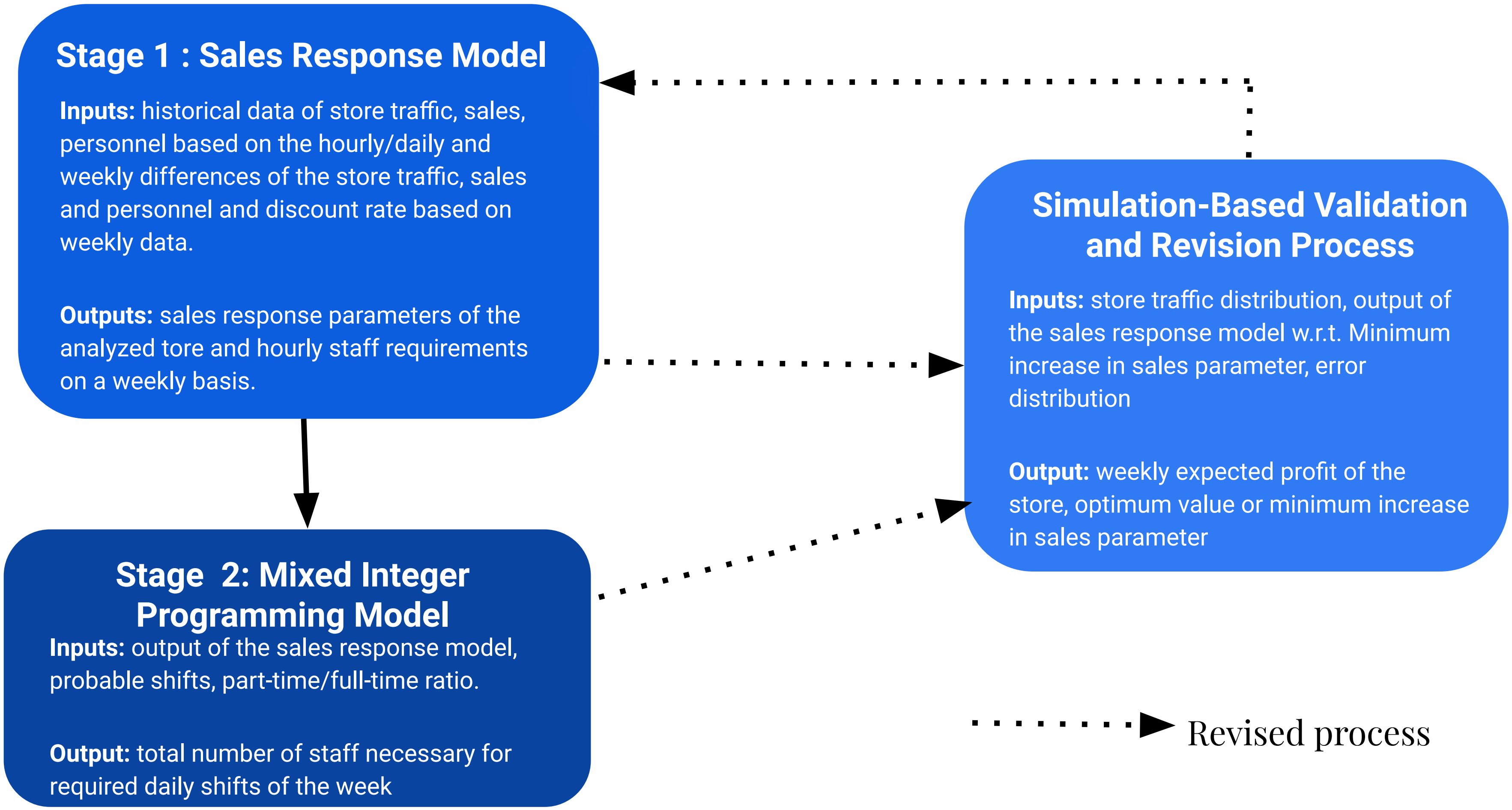
- Overstaffing leads to high labor cost.
- Understaffing would hurt the stores' profitability
- Inconsistency schedules for employees lower the staffs' satisfaction.
- Inconsistent customer service

Our Goal

Build effective and automated retail workforce scheduling models that :

- Ensure the number of staffs will match the store customer traffic
- Minimize the total staffing cost
- Improve the employees' satisfaction and wellbeing
- Help managers make better staff scheduling decisions based on staffing cost and the impact of staffing on the store revenues

Conceptual Framework of the Proposed Models



Stage 1-Sales Response Model

Assumptions: Not all customers react in the same manner to factors, such as the presence or absence of staff assistance, the level of in-store traffic, etc.

Input: store traffic, average price of the product, average discount on the store, number of salespeople

Output: expected sales (used as input for model 2)

$$N_t = \tau_m C_{t-1} + (1-\tau_m) C_t \quad \text{for } m = 0,1,2,3$$

$$S_t = \alpha_m N_t^{\beta_m} \mathrm{e}^{i_m \gamma_m / l_t} \quad \text{for } m = 0,1,2,3$$

$$\pi_t = S_t g - l_t d$$

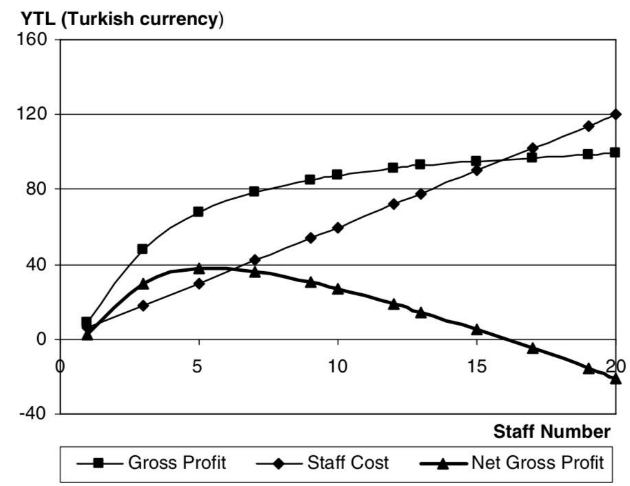


Fig. 2. Net gross profit, gross profit and staff cost functions as a function of staff number.

Stage 1 - Output

LC Waikiki, a Turkish retailer operating in the apparel sector, has three categories of stores based on their locations, sizes, and target population:

1. Neighborhood Business District (NBD)
2. **Shopping Mall (SM)**
3. Central Business District (CBD)

Output of Stage 1 Model/Input of Stage 2 Model

Hourly requirements of SM

Days/Hours	10	11	12	13	14	15	16	17	18	19	20	21
Monday	3	5	7	7	7	7	7	7	7	7	7	7
Tuesday	3	5	7	7	7	7	7	8	7	7	7	7
Wednesday	3	5	7	7	7	8	8	8	7	7	7	7
Thursday	3	5	7	7	7	7	7	7	7	7	7	7
Friday	3	5	7	7	7	8	8	8	7	7	7	7
Saturday	3	6	8	9	10	10	10	10	10	10	9	8
Sunday	3	6	8	10	10	11	11	11	10	10	9	8

Stage 2 - Original Version

$$\text{minimize } z = \text{CostF} * \sum_k \text{full}(k) + \text{CostP} * \sum_l \text{part}(l)$$

Subject to the following constraints:

$$\sum_l \text{part}(l) * (1 - \text{pfrate}) \leqslant \sum_k \text{full}(k) * \text{pfrate},$$

$$\text{FD} * \text{full}(k) \geqslant \sum_i \sum_j \text{assignfull}(i, j, k) \quad \forall k,$$

$$\text{PD} * \text{part}(l) \geqslant \sum_i \sum_i \text{assignpart}(i, j, l) \quad \forall l,$$

$$\sum_i \sum_j \text{dursh}(j) * \text{assignfull}(i, j, k) \leqslant \text{FG} \quad \forall k,$$

$$\sum_i \sum_j \text{dursh}(j) * \text{assignpart}(i, j, l) \leqslant \text{PG} \quad \forall l,$$

Objective: minimize total labor cost of part-time and full-time staffs

Constraints:

1. Define allowed ratio of part-time to full-time staff
2. Limit full-time staff working days and days off
3. Limit part-time staff working days and days off
4. Place upper bound on # of working hours/week (full-time staff)
5. Place upper bound on # of working hours/week (part-time staff)

Stage 2 - Original Version

$$\sum_j \text{assignfull}(i,j,k) \leqslant 1 \quad \forall k, i,$$

$$\sum_j \text{assignpart}(i,j,l) \leqslant 1 \quad \forall k, i,$$

$$\sum_s (\text{stafftosh}(s,i) * \text{workh}(s,h)) \geqslant \text{hreq}(i,h) \quad \forall i, h,$$

$$\sum_k \text{assignfull}(i,j,k) + \sum_l \text{assignpart}(i,j,l) \geqslant \sum_s (\text{stafftosh}(s,i) * \text{shiftint}(s,j)) \quad \forall i, j.$$

Constraints (cont):

1. Ensure full-time staff only assigned one shift per day
2. Ensure part-time staff only assigned one shift per day
3. Ensure part-time staff only assigned to one shift per day
   Guarantee that the hourly requirements resulting from the sales response are realized
4. The detailed shifts are transformed to hourly shifts per day, and the staff is assigned to those shifts

Stage 2 - Model Modifications

- Objective: Weekly salary => Hourly salary
- Output: assignment (day, shift length, employee) => assignment(person, shift schedule, employee)
- New constraints:
    - Minimum working days per week constraints
    - Minimum working hours per week constraints

Stage 2 - Model Improvements: “Employee Welfare” Constraints

- Original model produces an optimal scheduling for one week.
- However, **the welfare and satisfaction level associated with** a staff’s schedule from week to week are **NOT** considered.

Employee’s needs	***Solutions***
Staff prefer not always working on the weekends	Penalize when a staff has worked for 2 consecutive weekends
Staff prefer their shifts scheduling more consistent and predictable during the week.	Penalize the differences of the start time and end time of shifts assigned to a staff on the same weekday from week to week

Stage 2 - Model Improvement 1: Employees don't prefer working on weekends

$$Min \sum_{k,i,s} assign\_full(k,i,s) * dict\_cost\_full(k,i,s) + assign\_part(l,i,s) * dict\_cost\_part(l,i,s)$$
$$+WdPenaltyCost * (\sum_{k=1}^{num_{full}} owdfull(k) * \text{wdfull(k)} + \sum_{k=1}^{num_{part}} owdpart(l) * \text{wdpart(l)})$$
$$+CstPenaltyCost * (\sum_{k,i,s} start\_diff\_full(k,i,s) + end\_diff\_full(k,i,s) + start\_diff\_part(l,i,s) + end\_diff\_part(l,i,s))$$

$$\sum_{s=1}^{num\_shifts} \text{assignfull(k, 6, s)} \leq \text{wdfull(k)}, \forall k$$

$$\sum_{s=1}^{num\_shifts} \text{assignfull(k, 7, s)} \leq \text{wdfull(k)}, \forall k$$

$$\sum_{s=1}^{num\_shifts} \text{assignpart(l, 6, s)} \leq \text{wdfull(l)}, \forall l$$

$$\sum_{s=1}^{num\_shifts} \text{assignpart(l, 7, s)} \leq \text{wdfull(l)}, \forall l$$

1. Objective revised: Penalize the occurrences where a staff has worked for 2 consecutive weekends
2. Add 4 new constraints are to ensure that the indicator (whether a full-time/part-time staff work on weekend) is 1 if the staff works on any shifts on Saturday or Sunday
3. By multiplying last week's indicator and this week's indicator, we can track whether the staff has worked for 2 consecutive weeks

Stage 2 - Model Improvement 2: Employees prefer consistent scheduling

$$Min\ \sum_{k,i,s} assign\_full(k,i,s) * dict\_cost\_full(k,i,s) + assign\_part(l,i,s) * dict\_cost\_part(l,i,s)$$
$$+\text{WdPenaltyCost} * (\sum_{k=1}^{num_{full}} owdfull(k) * \text{wdfull(k)} + \sum_{k=1}^{num_{part}} owdpart(l) * \text{wdpart(l)})$$
$$+\text{CstPenaltyCost} * (\sum_{k,i,s} start\_diff\_full(k,i,s) + end\_diff\_full(k,i,s) + start\_diff\_part(l,i,s) + end\_diff\_part(l,i,s))$$

$$start\_diff\_full(k,i) \geq |\sum_{s=1}^{number\_shifts} \text{assignfull}_{(k,i,s)} * \text{starttime}_{(s)} - \sum_{s=1}^{num\_shifts} \text{oassignfull}(k,i,s) * \text{starttime}(s)| \quad \forall k,i$$

$$end\_diff\_full(k,i) \geq |\sum_{s=1}^{number\_shifts} \text{assignfull}_{(k,i,s)} * \text{endtime}_{(s)} - \sum_{s=1}^{number\_shifts} \text{oassignfull}(k,i,s) * \text{end\_time}(s)| \quad \forall k,i$$

$$start\_diff\_part(l,i) \geq |\sum_{s=1}^{number\_shifts} \text{assignpart}_{(l,i,s)} * \text{starttime}_{(s)} - \sum_{s=1}^{number\_shifts} \text{oassignpart}(l,i,s) * \text{starttime}(s)| \quad \forall k,i$$

$$end\_diff\_part(l,i) \geq |\sum_{s=1}^{number\_shifts} \text{assignpart}_{(l,i,s)} * \text{endtime}_{(s)} - \sum_{s=1}^{number\_shifts} \text{oassignfull}(l,i,s) * \text{endtime}(s)| \quad \forall k,i$$

1. Objective revised: Penalize based on how different are the shifts that are assigned to the same staff from week to week, by comparing their start time and end time

2. The four new constraints are to track start time differences and end time differences of the shifts, for the same staff on the same day of the week.

3. Eg. Monday (1st week): 9am-1pm. Monday (2nd week): 10am-3pm ↠ Differences = 3hrs

Stage 2 - Output

Objective without additional constraints: **$8700**

Objective with weekend and consistency constraints: **$11818**

Result: **35.84%** increase in the total cost

Week 1 - Original Model

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
10:00 - 11:00							
11:00 - 12:00							
12:00 - 13:00							
13:00 - 14:00							
14:00 - 15:00							
15:00 - 16:00							
16:00 - 17:00							
17:00 - 18:00							
18:00 - 19:00							
19:00 - 20:00							
20:00 - 21:00							
21:00 - 22:00							

Week 2 - Original Model

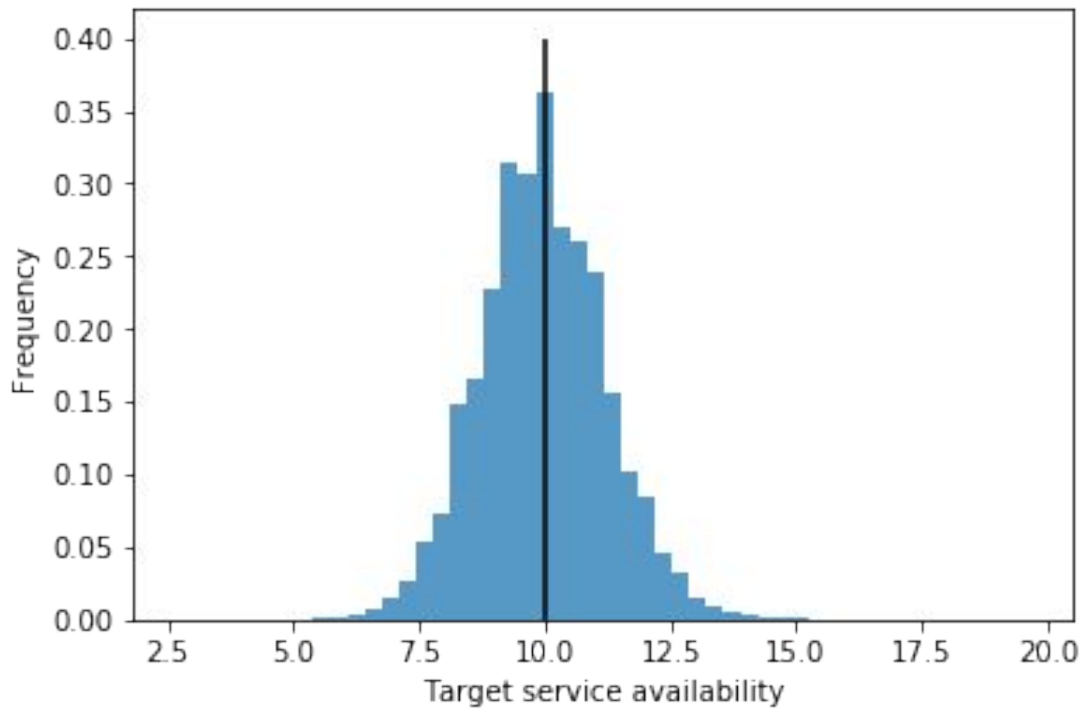
	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
10:00 - 11:00							
11:00 - 12:00							
12:00 - 13:00							
13:00 - 14:00							
14:00 - 15:00							
15:00 - 16:00							
16:00 - 17:00							
17:00 - 18:00							
18:00 - 19:00							
19:00 - 20:00							
20:00 - 21:00							
21:00 - 22:00							

Week 2 - Improved Model

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
10:00 - 11:00							
11:00 - 12:00							
12:00 - 13:00							
13:00 - 14:00							
14:00 - 15:00							
15:00 - 16:00							
16:00 - 17:00							
17:00 - 18:00							
18:00 - 19:00							
19:00 - 20:00							
20:00 - 21:00							
21:00 - 22:00							

Stage 2 - Probabilistic Constraints

Probability of not meeting target service level is **0.45** in a given hour and it's **1** in a given week.



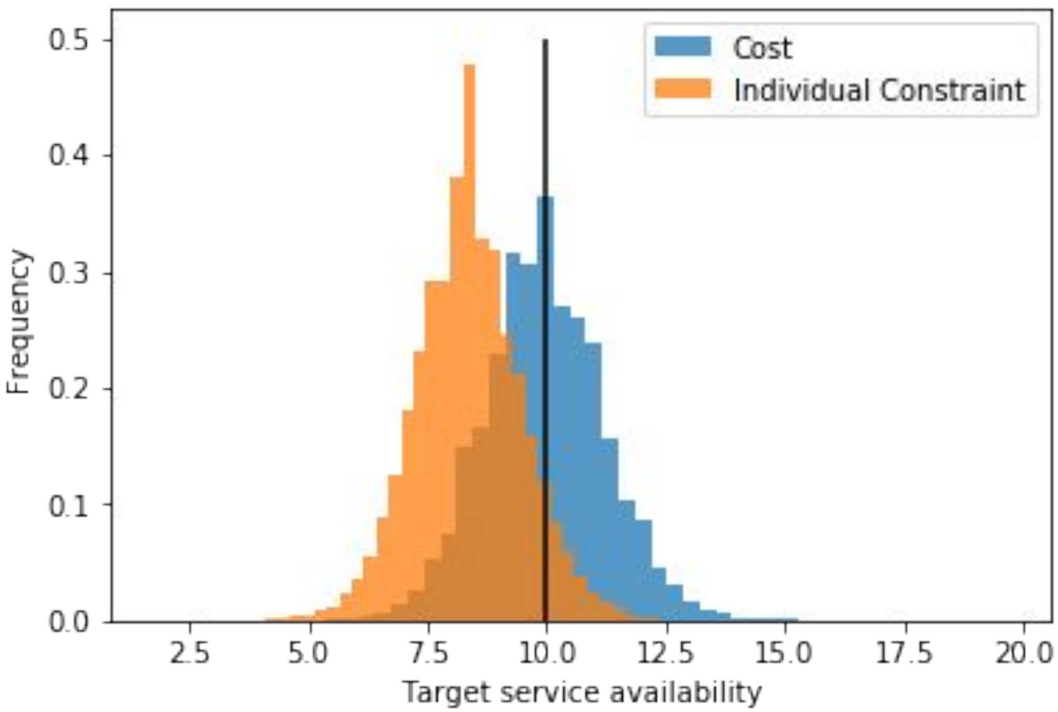
Minimize Cost with expected store traffic

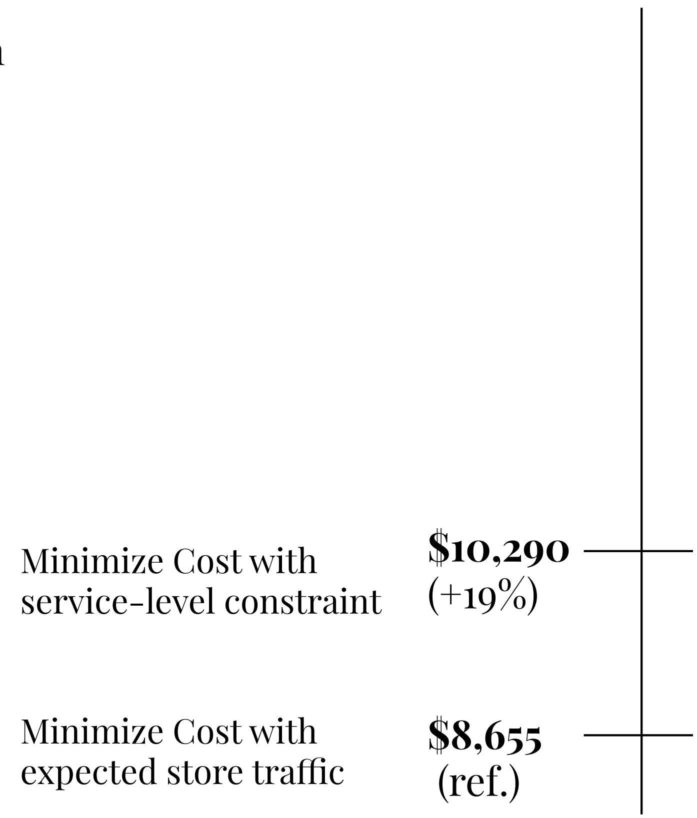
$8,655 (ref.)

Stage 2 - Probabilistic Constraints

$$\mathbb{P}[g_j(x, Z) \leq 0] \geq p_j, \; j \in J, \, p_j \in [0, 1]$$

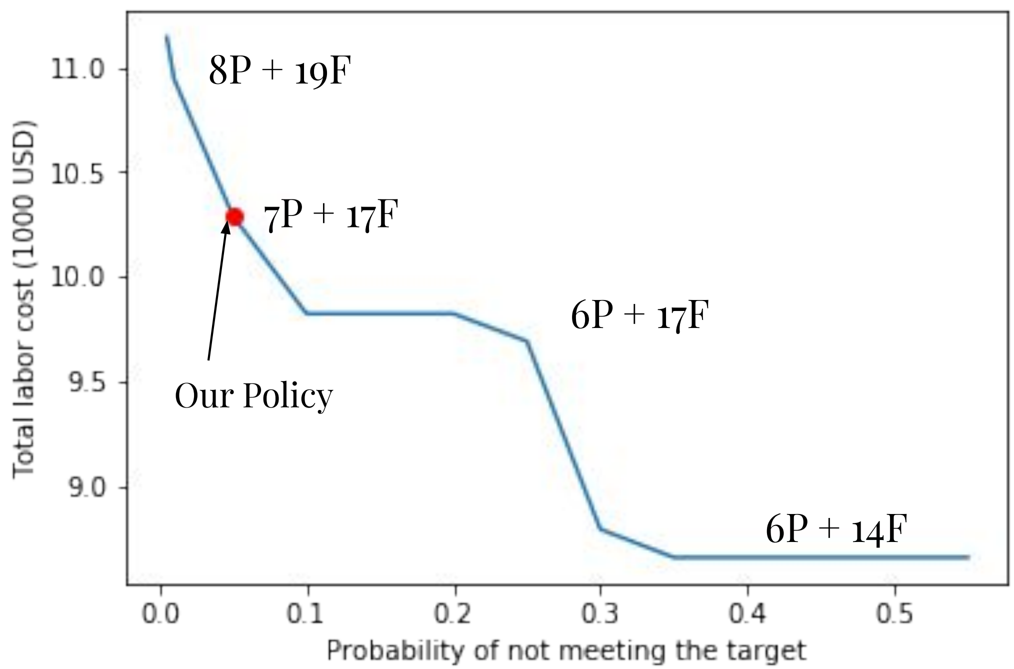
Probability of not meeting target service level is **0.06** in a given hour and it's **0.9** in a given week.





Stage 2 - Probabilistic Constraints

Probability of not meeting target service level is **0.06** in a given hour and it's **0.9** in a given week.



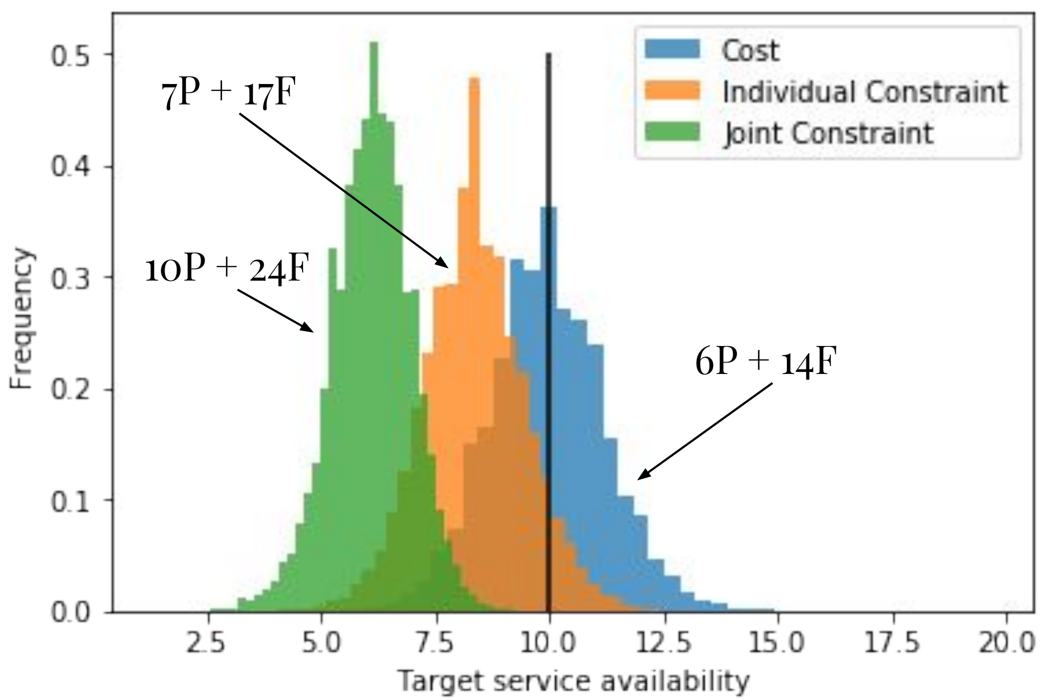
Minimize Cost with service-level constraint — **$10,290** (+19%)

Minimize Cost with expected store traffic — **$8,655** (ref.)

Stage 2 - Probabilistic Constraints

$$\mathbb{P}[g_j(x, Z) \le 0] \ge p_j,\ \ j \in J,\ p_j \in [0, 1]$$

Probability of not meeting target service level is **0.00001** in a given hour and it's **0.0013** in a given week.



Minimize Cost with stringent service-level constraint — **$13,395** (+54%)

Minimize Cost with service-level constraint — **$10,290** (+19%)

Minimize Cost with expected store traffic — **$8,655** (ref.)

Conclusions

- Matching labor to incoming traffic is a key driver of retail store profitability.
- The previous models show the tradeoffs that any retailer face between the cost of staffing, and welfare of its employees, and service level to its customers.
- Consistency in the schedule of a employee might increase his productivity and satisfaction and ultimately help drive the profitability of a company.
- Probability constraints are a very effective way to account for the uncertainty in the parameters and dynamics of the system. They provide a feasible region where we can reliably provide a service level and deliver on our value proposition.

References

[1] Lam, S., Vandenbosch, M., Pearce M. Retail Sales Force Scheduling Based on Store Traffic Forecasting. Journal of retailing.

[2] Kabak, O., Ulengin, F. et al. Efficient shift scheduling in the retail sector through two-stage optimization. European Journal of Operations Research.


Thank You