Optimization Assignment 3
Office Space Planning Optimization

Utpalraj Kemprai
MDS202352

1 Assumptions

1. **Flexible Seating Policy**: Employees do not have dedicated seats but can be assigned different seats on different days. This means that an employee from the Marketing team might sit at desk A on Monday and desk B on Tuesday.

2. **Social Distancing**: Floor capacities have been adjusted to accommodate social distancing requirements. For example, a floor that normally fits 100 people might now have a reduced capacity of 60 people.

3. **Collaboration Requirements**: Teams have specific days when collaboration is required, and team members should be allocated together on those days. For instance, if the Finance team needs to collaborate on Wednesdays, most team members should be in the office together on that day.

4. **Fixed vs. Floating Seats**: Some employees require fixed seats (guaranteed allocation), while others can use floating seats. Fixed seats are reserved for employees who must be in the office regularly, while floating seats can be used by any employee when needed.

5. **Cost Saving on Low Demand Days**: On days with low demand (e.g., near holidays), it's preferable to concentrate employees on fewer floors to reduce operational costs like heating, cleaning, and security.

6. **Night Shift Exclusion**: Night shift employees are excluded from the model as they can use the same spaces after day shift employees leave, effectively allowing the same physical seat to be used twice per day.

7. **Minimum Allocation Threshold**: When a team is allocated to a floor on a day, at least 5 employees must be assigned (or none at all). This prevents inefficient small clusters of team members.

8. **Business Hours**: All allocations are for standard business hours (typically 9am-5pm).

9. **Team Continuity**: Teams should ideally occupy the same floor(s) across different days of the week to provide consistency and familiarity for employees.

10. **Floor Preference**: Teams may have preferred floors for their operations based on proximity to resources, meeting rooms, or other amenities.

2 Sets

These are the categories and groupings used in our model:

- T: Set of teams, $T = \{1, 2, \ldots, |T|\}$ (e.g., Marketing, Finance, Engineering, etc.)
- D: Set of days in a week, $D = \{1, 2, \ldots, |D|\}$ (typically Monday through Friday)
- F: Set of floors, $F = \{1, 2, \ldots, |F|\}$ (the available floors in the building)
- C: Set of collaboration levels, $C = \{\text{High}, \text{Medium}, \text{Low}\}$ (how much a team needs to work together)

3 Parameters

These are the fixed inputs to our model:

- C_f: Capacity of floor $f \in F$ (adjusted for social distancing)
- E_t: Total number of employees in team $t \in T$ (the size of each team)
- F_t: Number of fixed seats required by team $t \in T$ (guaranteed spots needed)
- P_{td}: Collaboration pattern for team $t \in T$ on day $d \in D$ (1 if collaboration is required, 0 otherwise)
- $L_t \in C$: Collaboration level needed by team $t \in T$ (high, medium, or low)
- $\text{demand}_d \in [0, 1]$: Demand for day d. Gives a score between 0 and 1 to quantify demand on day d, higher the score corresponds to a higher demand.
- minAlloc: Minimum number of employees allocated per team per day per floor (= 5)
- X: Maximum loading percentage deviation (controls how balanced office occupancy is across days)
- w_1, w_2, w_3, w_4, w_5: Weights for different objectives in the objective function (importance of each goal)
- α: Minimum proportion of target collaboration to be satisfied (= 0.6 or 60%)
- Pref_{tf}: Preference value for team t to be assigned to floor f (higher value means higher preference)

4 Decision Variables

These are the values our optimization model will determine:

- $x_{tdf} \in \mathbb{Z}_{\geq 0}$: Number of employees from team $t \in T$ assigned to floor $f \in F$ on day $d \in D$
- $y_{tf} \in \{0, 1\}$: Binary variable, 1 if team $t \in T$ occupies floor $f \in F$, 0 otherwise
- $z_{td} \in \{0, 1\}$: Binary variable, 1 if any member of team $t \in T$ is present on day $d \in D$, 0 otherwise
- $w_t \in \{0, 1\}$: Binary variable, 1 if team t occupies more than one floor, 0 otherwise
- $\text{floor}_{df} \in \{0, 1\}$: Binary variable, 1 if floor f is used on day d, 0 otherwise
- minLoad: Minimum total loading across all days (lowest daily occupancy)
- maxLoad: Maximum total loading across all days (highest daily occupancy)
- $a_{tdf} \in \{0, 1\}$: Binary variable, 1 if team t has a non-zero allocation on floor f on day d, 0 otherwise
- $\text{dayShift}_{td} \in \{0, 1\}$: Binary variable, 1 if team t needs to shift collaboration to a different day than preferred (day d)
- $\text{floorDiff}_t \in \mathbb{Z}_{\geq 0}$: Number of different floor configurations team t uses across the week
- $u_{tdd'f} \in \{0, 1\}$: Binary variable for linearization, 1 if team t uses floor f on day d but not on day d'

5 Objective Function

$$\begin{aligned}\text{Maximize: } & w_1 \cdot \sum_{t \in T} \sum_{d \in D} \sum_{f \in F} x_{tdf} - w_2 \cdot \sum_{t \in T} w_t - w_3 \cdot \sum_{d \in D} \sum_{f \in F} \text{floor}_{df} \cdot (1 - \text{demand}_d) \\ & - w_4 \cdot (\text{maxLoad} - \text{minLoad}) - w_5 \cdot \sum_{t \in T} \text{floorDiff}_t + \sum_{t \in T} \sum_{f \in F} \text{Pref}_{tf} \cdot y_{tf} \end{aligned} \tag{1}$$

Explanation: The objective function represents what we're trying to achieve with our office space plan. It balances multiple business goals by assigning different weights to each goal:

1. **Maximize total occupancy** (w_1 term): Get as many employees into the office as possible. This ensures we're making good use of our office space investment.

   For example, if we have 500 total employees and can fit 400 in the office with social distancing, a higher w_1 value will push toward getting close to that 400 number.

2. **Minimize teams splitting across multiple floors** (w_2 term): Keep teams together for better collaboration. When teams are split across floors, communication becomes harder and team cohesion can suffer.

   For example, if the Marketing team has 30 people, we'd prefer all 30 to be on the same floor rather than 15 on Floor 2 and 15 on Floor 3.

3. **Minimize floors used on low-demand days** (w_3 term): Concentrate employees on fewer floors on quieter days (like before holidays) to save on operational costs.

   For example, if Friday before a holiday typically has only 60% attendance, we might close Floors 3 and 4 completely and only operate Floors 1 and 2, saving on utilities and services.

4. **Balance workload across days** (w_4 term): Avoid having some days extremely crowded and others nearly empty. This helps with consistent resource planning.

   For example, instead of having 400 employees on Monday and 200 on Friday, we might aim for a more balanced 320 employees each day.

5. **Minimize floor configuration changes** (w_5 term): Ensure teams occupy the same floors throughout the week. This reduces confusion and helps employees establish routines.

   For example, the Finance team should be on Floor 2 every day they're in the office, rather than Floor 2 on Monday and Floor 3 on Tuesday.

6. **Respect floor preferences**: Assign teams to their preferred floors where possible. Some teams may need to be close to certain resources or facilities.

   For example, the Executive team might prefer Floor 10 due to its proximity to the executive meeting rooms, while IT might prefer Floor 1 to be close to the server room.

Setting Priorities: By adjusting the weights w_1 through w_5, management can set priorities. For example, if we set $w_1 = 10, w_2 = 5, w_3 = 3, w_4 = 2, w_5 = 4$, the model will prioritize:

- First priority: Maximizing how many employees come to the office
- Second priority: Keeping teams on the same floors throughout the week
- Third priority: Keeping teams together on a single floor
- Fourth priority: Minimizing costs on low-demand days
- Fifth priority: Balancing attendance across the week

6 Constraints

6.1 Floor Capacity

$$\sum_{t \in T} x_{tdf} \leq C_f \cdot \text{floor}_{df}, \quad \forall d \in D, f \in F \tag{2}$$

Purpose: Ensures we don't exceed the maximum capacity of each floor.

Explanation: Think of this as a safety rule. Each floor has a maximum capacity that accounts for social distancing requirements. This constraint makes sure we never put more people on a floor than it can safely hold.

Example: If Floor 3 has a capacity of 50 people with social distancing, this constraint ensures we assign at most 50 people to that floor on Monday. If we try to assign 52 people (say, 30 from Marketing and 22 from Finance), the model will flag this as impossible and find a different solution, such as moving some Finance team members to another floor or having them work from home that day.

6.2 Fixed Seat Allocation

$$\sum_{d \in D} \sum_{f \in F} x_{tdf} \geq F_t, \quad \forall t \in T \tag{3}$$

Purpose: Guarantees that teams receive their required fixed seats.

Explanation: Some employees need guaranteed office access due to their roles. This constraint ensures each team gets at least their minimum required number of "fixed seats" across the week.

Example: If the Finance team requires 10 fixed seats (perhaps for employees who handle confidential information), this constraint guarantees at least 10 Finance employees will be allocated office space during the week. Without this constraint, the model might prioritize other teams and leave Finance with insufficient space, disrupting critical operations.

6.3 Collaboration Pattern

$$\sum_{f \in F} x_{tdf} \geq \alpha \cdot P_{td} \cdot E_t \cdot (1 - \text{dayShift}_{td}), \quad \forall t \in T, d \in D \tag{4}$$

Purpose: Respects each team's collaboration needs on specific days.

Explanation: Teams often have designated "collaboration days" when most team members need to be in the office together for meetings, brainstorming, or project work. This constraint ensures that on those days, at least a minimum percentage of the team is physically present, unless the collaboration day needs to be shifted.

Example: Let's say the Marketing team has 20 employees and needs to collaborate on Wednesdays ($P_{Marketing,Wednesday} = 1$). With $\alpha = 0.6$ (60%), this constraint ensures at least 12 Marketing employees (60% of 20) are in the office on Wednesday.

If there isn't enough space on Wednesday, the model might set $\text{dayShift}_{\text{Marketing,Wed}} = 1$, which would relax this constraint for Wednesday but require the team to collaborate on another day instead.

6.4 Collaboration Shift Limit

$$\sum_{d \in D} \text{dayShift}_{td} \leq 1, \quad \forall t \in T \tag{5}$$

Purpose: Limits the number of days that collaboration can be shifted for each team.

Explanation: While we want to accommodate all teams' collaboration needs, sometimes space constraints require flexibility. This constraint allows each team's collaboration day to be shifted at most once per week, ensuring teams can still plan their collaborative work effectively.

Example: If the HR team normally collaborates on Monday, but there's not enough space that day, the model might shift their collaboration to Tuesday. However, if they also collaborate on Thursday, this constraint prevents that second session from being shifted to another day. This gives teams predictability while allowing some flexibility.

6.5 Total Employee Allocation

$$\sum_{d \in D} \sum_{f \in F} x_{tdf} \leq |D| \cdot E_t, \quad \forall t \in T \tag{6}$$

Purpose: Prevents assigning more employee-days than a team actually has available.

Explanation: This is a reality check constraint. We can't assign more employee office days than we actually have employees. This ensures our plan stays grounded in reality and doesn't "double-count" employees.

Example: If the IT team has 30 employees total and we're planning for a 5-day work week, the maximum possible allocation is 30 employees × 5 days = 150 employee-days. This constraint prevents the model from assigning, for example, 35 IT employees on Monday (which is impossible since there are only 30 team members).

6.6 Daily Employee Limit

$$\sum_{f \in F} x_{tdf} \leq E_t, \quad \forall t \in T, d \in D \tag{7}$$

Purpose: Ensures we don't assign more employees from a team than exist on any given day.

Explanation: This constraint ensures that on any single day, we don't schedule more employees from a team than actually exist. While the previous constraint deals with the week as a whole, this one focuses on daily limits.

Example: If the Sales team has 25 members total, this constraint prevents us from assigning, for instance, 15 Sales team members to Floor 2 and 15 more to Floor 5 on Tuesday (which would total 30, exceeding the actual team size of 25). The model would instead limit the total to at most 25 Sales team members across all floors on any given day.

6.7 Team Floor Occupancy

$$x_{tdf} \leq E_t \cdot y_{tf}, \quad \forall t \in T, d \in D, f \in F \tag{8}$$

Purpose: Tracks which floors are occupied by each team.

Explanation: This constraint helps us keep track of which floors each team uses during the week. If any team member is assigned to a floor at any point during the week, we mark that floor as "occupied" by that team, which helps with planning team locations.

Example: If 15 Sales team members are assigned to Floor 2 on Monday, then $y_{Sales,2} = 1$, indicating that the Sales team occupies Floor 2 at some point during the week. If no Sales team members are ever assigned to Floor 3 during the entire week, then $y_{Sales,3} = 0$. This tracking helps us analyze team distribution patterns across the building.

6.8 Team Presence on Day

$$x_{tdf} \leq E_t \cdot z_{td}, \quad \forall t \in T, d \in D, f \in F \tag{9}$$

Purpose: Tracks which days each team is present in the office.

Explanation: Similar to the floor occupancy tracking, this constraint helps us monitor which days each team comes into the office. This information is useful for planning support services, catering, and other day-specific resources.

Example: If any Engineering team members are assigned to any floor on Tuesday, then $z_{Engineering,Tuesday} = 1$, indicating the Engineering team is present on Tuesday. If no Engineering team members are in the office on Friday, then $z_{Engineering,Friday} = 0$. These indicators help management see team attendance patterns across the week.

6.9 Loading Balance

$$\sum_{t \in T} \sum_{f \in F} x_{tdf} \geq \text{minLoad}, \quad \forall d \in D \tag{10}$$

$$\sum_{t \in T} \sum_{f \in F} x_{tdf} \leq \text{maxLoad}, \quad \forall d \in D \tag{11}$$

$$\text{maxLoad} \leq (1 + X) \cdot \text{minLoad} \tag{12}$$

Purpose: Ensures balanced occupancy across different days of the week.

Explanation: These constraints prevent extreme variations in daily office occupancy. Having some days extremely crowded and others nearly empty makes planning difficult and creates inefficiencies. These constraints ensure that the busiest day isn't overloaded compared to the quietest day.

Example: If we set $X = 0.2$ (20%) and the model determines that the minimum daily occupancy (minLoad) is 100 employees (perhaps on Friday), then the maximum daily occupancy (maxLoad) cannot exceed 120 employees (on any day of the week). This means if Monday would naturally attract 150 employees, the model will need to reassign some of them to other days to maintain balance, perhaps by shifting some team collaboration days.

6.10 Minimum Allocation

$$x_{tdf} \geq \text{minAlloc} \cdot a_{tdf}, \quad \forall t \in T, d \in D, f \in F \tag{13}$$

$$x_{tdf} \leq E_t \cdot a_{tdf}, \quad \forall t \in T, d \in D, f \in F \tag{14}$$

Purpose: Enforces a minimum threshold for team members on a floor.

Explanation: Having just one or two team members on a floor is often inefficient. This constraint ensures that when a team is assigned to a floor, either none of its members are assigned there (zero allocation) or at least a minimum number of team members (e.g., 5 employees) are assigned together.

Example: If the minimum allocation (minAlloc) is 5 employees, this constraint prevents scenarios like having just 3 HR team members on Floor 1 on Monday. Either 0 HR members or at least 5 HR members must be assigned to Floor 1 on Monday. This promotes more efficient clustering of team members and prevents isolated small groups that might struggle with collaboration.

6.11 Multiple Floors per Team

$$\sum_{f \in F} y_{tf} - 1 \leq |F| \cdot w_t, \quad \forall t \in T \tag{15}$$

Purpose: Identifies teams that must be split across multiple floors.

Explanation: This constraint tracks which teams need to be split across multiple floors. Since team splitting is generally undesirable (it makes collaboration harder), the objective function penalizes this situation.

Example: If the Legal team needs to use both Floor 2 and Floor 3 during the week (perhaps because the team is too large to fit on a single floor), then $w_{Legal} = 1$, indicating the team is split across floors. This activates a penalty in the objective function, encouraging the model to find solutions that keep teams together when possible. If Legal only uses Floor 2, then $w_{Legal} = 0$ (no penalty).

6.12 Teams with High Collaboration Needs on Same Floor

$$\sum_{f \in F} y_{tf} = 1, \quad \forall t \in T \text{ where } L_t = \text{High} \tag{16}$$

Purpose: Keeps high-collaboration teams on a single floor.

Explanation: Teams that require intensive collaboration need to be physically close together. This constraint ensures that teams marked as having ”High” collaboration needs are always assigned to exactly one floor, never split across multiple floors.

Example: If the Product Development team has a high collaboration level ($L_{ProductDev} =$ High), they will be restricted to occupying only one floor, even if it means some team members might not get office space on certain days. For instance, if Product Development has 40 members but the floor capacity is only 35, some team members might need to work from home rather than splitting the team across multiple floors.

6.13 Fair Share Allocation

For floating seats (after fixed allocation):

$$\sum_{d \in D} \sum_{f \in F} (x_{tdf} - F_t \cdot z_{td}) \geq \frac{E_t - F_t}{\sum_{t' \in T} (E_{t'} - F_{t'})} \cdot \text{TotalFloatingSeats}, \quad \forall t \in T \tag{17}$$

where $\text{TotalFloatingSeats} = \sum_{f \in F} C_f - \sum_{t \in T} F_t$

Purpose: Ensures equitable distribution of floating seats across teams.

Explanation: After all fixed seat requirements are met, the remaining ”floating” seats should be distributed fairly across teams based on their size. This constraint ensures each team gets at least their proportional share of the flexible office space.

Example: Let's say after assigning all fixed seats, there are 100 floating seats available. If the Marketing team has 20 employees (of which 5 need fixed seats) and represents 15% of the remaining employee population needing seats, they should receive at least 15 floating seats across the week. Without this constraint, some teams might get more than their fair share of floating seats while others get less.

6.14 Floor Configuration Consistency

To linearize these constraints, we introduce auxiliary binary variables $u_{tdd'f}$ which indicate if team t uses floor f on day d but not on day d'.

$$z_{td} \cdot y_{tf} - z_{td'} \cdot y_{tf} \leq u_{tdd'f}, \quad \forall t \in T, d, d' \in D, d \neq d', f \in F \tag{18}$$

$$z_{td'} \cdot y_{tf} - z_{td} \cdot y_{tf} \leq u_{td'd,f}, \quad \forall t \in T, d, d' \in D, d \neq d', f \in F \tag{19}$$

$$\sum_{d \in D} \sum_{d' \in D, d' \neq d} \sum_{f \in F} u_{tdd'f} \leq \text{floorDiff}_t, \quad \forall t \in T \tag{20}$$

Purpose: Measures how often a team changes its floor configuration during the week.

Explanation: Teams prefer consistency in their floor allocation throughout the week. These constraints track how many times a team's floor configuration changes from day to day, and the objective function penalizes frequent changes.

Example: If the Finance team uses Floor 1 on Monday and Tuesday, but then uses both Floors 1 and 2 on Wednesday, this counts as one configuration change ($\text{floorDiff}_{Finance} = 1$). If they then switch to just Floor 2 on Thursday, that's another change, so $\text{floorDiff}_{Finance} = 2$. The objective function will penalize these changes, encouraging the model to find solutions where teams use consistent floor allocations throughout the week.

7 Model Output

The model provides the following output:

1. **Detailed Allocation Plan**: Number of employees from each team assigned to each floor on each day (x_{tdf}).

   For example: "15 Marketing team members on Floor 3 on Monday, 22 on Tuesday..."

2. **Team Floor Occupancy**: Which floors are occupied by each team (y_{tf}).

   For example: "The Sales team occupies Floors 2 and 3. The IT team occupies Floor 1 only."

3. **Daily Team Presence**: Which days each team has employees in the office (z_{td}).

   For example: "The Engineering team is present Monday, Tuesday, and Thursday."

4. **Floor Usage**: Which floors are in use on which days (floor_{df}).

   For example: "On Friday, only Floors 1 and 2 are in use; Floors 3 and 4 are closed."

5. **Compliance with Collaboration Requirements**: Whether teams are meeting their collaboration needs on specified days or if shifts were needed.

   For example: "The HR team's collaboration was shifted from Monday to Wednesday due to space constraints."

6. **Fairness Metrics**: How equitably the floating seats are distributed across teams.

   For example: "The Finance team represents 18% of employees needing floating seats and received 19% of available floating seats."

7. **Efficiency Metrics**: How many teams need to split across multiple floors and how many floor configuration changes occur during the week.

   For example: "3 out of 10 teams need to split across multiple floors. The average team changes floor configuration 1.2 times per week."

These outputs allow management to implement a flexible seating arrangement that maximizes office space utilization while respecting team collaboration needs and maintaining fairness in allocation. The detailed plan can be shared with employees through the company intranet or scheduling system so everyone knows which floors they should report to each day.