Project Overview: Periodic Scheduling of Student-Professor Meetings

Aradhya Jindal, Yashaswini Mathur

April 23, 2025

Outline

Problem Statement

We consider a scheduling problem where students from different academic groups (Undergraduate, Master's, and PhD) are required to meet with a set of professors over a fixed number of days and time slots. Each group has its own visit requirements and constraints. The goal is to assign student visits to professors in a way that minimizes a cost function subject to practical scheduling constraints.

For example some key practical constraints include:

- Minimum intervals between consecutive meetings. (B)
- Varying time requirements for different student levels (undergraduate, master's, PhD). (T)
- Dedicated emergency slots for unplanned visits. (E)

Our Work

Some features that we introduce in our solution over the one in [1]:

- Give cleaner formulation by introducing W_x, W_y and W_z.
- Introduce priority to the categories of students. For example, if meetings with the phd take priority over undergrads then their meetings are scheduled in the first available slot over undergrads.
- Expand the current model to multiple service providers with common customers.
- Provide a modification to the objective function that prioritizes earlier days for efficiency.

Why Our Work is Important?

- **Designed for Multi-Provider Environments** – Unlike many existing systems that cater to individual providers, our framework supports scheduling across multiple professionals with shared or distributed responsibilities.
- **Promoting Fairness and Operational Efficiency** – The model ensures a balanced allocation of appointments among providers while reducing scheduling conflicts and maximizing utilization of available time slots.
- **Incorporating Flexibility for Emergencies** – Real-world scenarios often involve last-minute changes. Our approach integrates emergency slots into the schedule to accommodate urgent requests without compromising overall stability.
- **Foundation for Intelligent Scheduling** – This framework lays the groundwork for future integration with machine learning techniques to enable predictive scheduling, adaptive slot adjustments, and data-driven optimization.

Overview of Model Formulation

In this framework, we model the scheduling problem where **students are treated as customers**, and **faculty members as service providers**.
Each professor's daily availability is divided into P half-hour time slots.
Students are categorized into three groups: Undergraduate ($s = x$), Master's ($s = y$), PhD ($s = z$).
Each student type has distinct requirements for meetings:

- Type-s students require R_s consecutive half-hours per meeting.
- They must schedule T_s meetings within the planning period.
- A minimum of B_s days must separate any two consecutive meetings.
- Have a priority weight of M_s.
- Have a map p_s, that specifies the profs a student of type s has to meet with.

In our simulations, each student has identical meeting requirements across all professors, though this can be easily customized with minor code modifications.

Indices used in the model

- $i = 1, \ldots, N_s$: student index (by level)
- $j = 1, \ldots, D$: day index (D is number of planning days)
- $k = 1, \ldots, P$: half-hour slot index (P is the number of half-hours slots for all profs in a day)
- $p = 1, \ldots, p^s[i]$: Professors that student i of type s needs to meet.

Parameter used in the model

- N_x, N_y, N_z: # undergrad, master, PhD students
- T_x, T_y, T_z: required visits per student
- R_x, R_y, R_z: slots per visit
- B_x, B_y, B_z: minimum day-gap
- M_x, M_y, M_z: Weights for each category of students
- L: emergency slots/day
- P: number of half-hours slots for all profs in a day
- D: number of planning days
- Q: Total number professors
- F_{jk}: availability (1 or 0)
- p^s: $[[0],[1]] \rightarrow$ First student of type s wants to meet 1st prof and second student of type s wants to meet 2nd prof.

Decision Variables

- $X_{i,j,k,p} = 1$ if undergrad i visits prof p on day j slot k.
- $Y_{i,j,k,p} = 1$ for master students.
- $Z_{i,j,k,p} = 1$ for PhD students.
- $E_{j,k,p} = 1$ if slot (j, k) is for emergency visits for professor p.
- $W^x_{i,j,p} = 1$ if student i of type x has any visit on day j with prof p.
- $W^y_{i,j,p} = 1$ if student i of type y has any visit on day j with prof p.
- $W^z_{i,j,p} = 1$ if student i of type z has any visit on day j with prof p.

Objective Function

$$\min \; 2\sum_{i,j,k}(jP+k+1)\Big(M_x\sum_{p^x} X_{i,j,k,p} + \; M_y\sum_{p^y} Y_{i,j,k,p} + \; M_z\sum_{p^z} Z_{i,j,k,p}\Big)$$

$$+ (jP+k+1)\sum_{j,k}\sum_{p\in[Q]} E_{j,k,p}$$

- **Student groups:** $X_{ijkp}, Y_{ijkp}, Z_{ijkp}$ capture under-grad, master's and PhD visits, scaled by weights M_x, M_y, M_z so that we can prioritise one cohort over another.
- **Time-of-day penalty:** The factor $(jP + k + 1)$ is a *global slot index*: Pushes meetings to the earliest feasible day/slot and spreads the load.
- **Regular vs emergency:** The outer coefficient 2 doubles the cost of *planned* visits, while emergency slots E_{jkp} appear without it. The result of this is that emergencies are only scheduled once all cheaper regular visits are placed, so they naturally settle into the tail of each day.

Emergency Slots

$$\sum_{k,p} E_{j,k,p} = L \quad \forall\, j \in [D],\ \forall\, p \in [Q]$$

- This constraint ensures that each day j, for each professor p, exactly L time slots are reserved for emergency or unplanned meetings.
- Emergency slots provide flexibility to accommodate last-minute student needs without disrupting the regular schedule.
- These slots are typically allocated after all planned appointments to minimize interference with pre-scheduled meetings.

Capacity Constraints

$$\sum_{i}^{N_x} X_{i,j,k,p} + \sum_{i}^{N_y} Y_{i,j,k,p} + \sum_{i}^{N_z} Z_{i,j,k,p} + E_{j,k,p} \leq F_{j,k,p}$$
$$\forall\ j \in [D],\ \forall\ k \in [P],\ \forall\ p \in [Q]$$

- This constraint ensures that no time slot is double-booked for any professor.
- The sum includes all students (undergraduate, master's, PhD) assigned to professor p at time (j, k), along with emergency bookings.
- The total cannot exceed the availability matrix $F_{p,j,k}$, which encodes whether professor p is available at that time.

Allocation Rules

- **Meeting Duration per Day:**

$$\sum_{k=0}^{P-1} X_{i,j,k,p} = R_x \cdot W^x_{i,j,p} \quad \forall i \in [N_x],\ \forall p \in p^x_i,\ \forall j \in [D]$$

  Ensures each meeting spans exactly R_x consecutive slots if scheduled on day j.

- **No Overlapping Meetings:**

$$X_{i,j,k,p} + X_{i,j,h,p} \leq 1 \quad \forall k, h \in [P],\ h \geq k + R_x,\ \forall i, j, p$$

  Prevents overlapping multiple meeting blocks on the same day.

Similarly describe the same for Y and Z.

Total meetings and gaps

- **Total Meeting Count:**

$$\sum_{j=0}^{D-1} \sum_{k=0}^{P-1} X_{i,j,k,p} = R_x \cdot T_x \quad \forall i \in [N_x],\ \forall p \in p_i^x$$

  Guarantees the student has T_x meetings in total.

- **Minimum Gap Between Meetings:**

$$\sum_{j=t}^{t+B_x-1} \sum_{k=0}^{P-1} X_{i,j,k,p} \leq R_x \quad \forall i \in [N_x],\ \forall p \in p_i^x,\ \forall t \in [0, D - B_x)$$

  Prevents scheduling another meeting within B_x days of a previous one.

Similarly describe the same for Y and Z.

Small Example Parameters

Type	R	N	T	B	M
Undergrad	1	4	3	1	1
Master	2	5	2	2	2
PhD	3	4	1	3	4
Days D			7		
Slots P			6		
Empties L			1		
Profs Q			4		

Professor Availability Matrix (F): In this example each professor is available in all time slots: $F[q][j][k] = 1$ for all $q \in [Q], j \in [D], k \in [P]$

Student-Professor Assignments:

- $p\_x = [[0,1],[1,2,3],[2,4],[3]]$ *(Undergraduates)*
- $p\_y = [[1,2],[2],[3],[0,3],[0,4]]$ *(Master's)*
- $p\_z = [[2,4],[0,3],[0,1,2],[3,4]]$ *(PhD)*

Small Example Results

- Constraints: 2244
- Variables: 1344
- Objective: 4324

Table: Schedule for Professor 1

	1	2	3	4	5	6
Monday	P2	P2	P2	M4	M4	E
Tuesday	P4	P4	P4	M5	M5	E
Wednesday	P3	P3	P3	M4	M4	E
Thursday	M5	M5	U1	E		
Friday	U1	E				
Monday	U1	E				
Tuesday	E					

Conclusion & Future Work

- The ILP model effectively schedules periodic student-professor meetings, ensuring efficiency while handling practical constraints like time gaps and emergency slots.
- It provides a balanced distribution of appointments across professors, with built-in mechanisms for high-priority students and time-sensitive meetings.
- Future Work:
  - Dynamic Real-Time Updates:
  - Integration of Machine Learning
  - **S**calability
  - **D**evelop User-Friendly Interface

References

1 **Main paper:** An Integer Linear Program for Periodic Scheduling in Universities: https://arxiv.org/abs/2412.11941

2 Smith, J. (2023). *Optimization Techniques for Academic Scheduling*. Springer.

3 Doe, A., Lee, B. (2022). "A Comparative Study of Scheduling Algorithms in Educational Institutions." *Journal of Operations Research*, 45(3), 215-230.

4 Link to code: https://github.com/Picozen2718/An-ILP-for-Periodic-Scheduling.git