A Mixed-Integer Programming Framework for Optimal TA Tutorial Assignments

Xander Pauwels

March 27, 2025

Overleaf project link

Abstract

In this report, we investigate a tutorial assignment problem arising from the need to allocate teaching assistants (TAs) to multiple tutorial sessions. Each TA has limited availability, specific time-slot preferences, and is permitted to lead no more than two tutorials. We formulate the problem as a mixed-integer linear program that ensures every session is adequately staffed while maximizing the alignment between TA preferences and their assigned sessions. To verify the robustness of our model, we perform sensitivity checks by varying TA availability and preference parameters to assess how changes affect the optimal assignment. The optimal assignment derived from our model significantly outperforms a manually constructed alternative in terms of preference satisfaction, even under these perturbed scenarios.

1 Introduction

Tutorial scheduling is a specific instance of the broader workforce scheduling problem, where employees or contractors are assigned to shifts while balancing their availability and preferences. Foundational reviews by Van den Bergh et al. (2013) and Brucker et al. (2011) provide comprehensive overviews of personnel scheduling challenges and solution strategies. These studies detail the constraints and computational complexities that emerge when managing conflicting requirements, a critical consideration when assigning teaching assistants (TAs) to tutorial sessions.

In parallel, domain-specific research offers further insights into managing complex scheduling issues. For instance, Dowsland (1998) demonstrates how heuristic methods such as tabu search applied to nurse rostering can effectively handle intricate scheduling constraints. More directly aligned with our study, Qu et al. (2017) propose mixed-integer linear programming (MILP) models specifically for TA assignments, addressing challenges related to limited availability and specific time-slot preferences. Complementing these approaches, Bixby (2012) describes the evolution and computational strengths of MILP techniques, underscoring their suitability for solving complex staffing and timetabling problems. By synthesizing these contributions, our work develops an MILP framework that not only ensures adequate tutorial coverage but also significantly enhances TA preference satisfaction, addressing the unique challenges posed by TAs who must balance their teaching duties with their own academic commitments.

Our setting involves a large undergraduate mathematics course at a university, in which numerous TAs each lead one or two tutorials. Because TAs are students themselves, they have their own class timetables, hence the need to respect strict availability constraints. Further, TAs often have personal or academic preferences for particular slots. Accounting for these factors complicates the scheduling problem and can lead to suboptimal staff satisfaction if done manually. Constructing a schedule by hand is both error-prone and time-consuming, particularly when seeking an assignment that simultaneously covers required time slots and maximizes preference satisfaction. In response to these challenges, this paper contributes a novel mixed integer linear programming (MILP) formulation that explicitly integrates both the coverage requirements and the individual constraints of TAs. Our model leverages techniques in integer programming to generate an optimized schedule that improves the alignment between TA preferences and their assigned slots, compared to an existing manually generated solution. The empirical results demonstrate that our approach significantly enhances preference satisfaction.

The remainder of this paper is organized as follows. Section 2 describes the scheduling problem, the input parameters, and the constraints in detail. Section 3 provides our linear integer programming formulation and explains the solution approach via Python and PuLP. Section 4 presents the numerical experiments, including the comparison to the handcrafted schedule. Finally, Section 5 summarizes the main findings and suggests directions for future research.

2 Problem Description

We consider a tutorial scheduling problem for a first-year mathematics course, where teaching assistants are required to lead multiple tutorial sessions. Each session takes place in a predefined set of time slots, and each time slot must be staffed by exactly g_i TAs to accommodate all student tutorial groups. However, as TAs themselves are full-time students, their availability for tutorial sessions is limited by their own class schedules; specifically, a TA j can only be assigned to teach in a time slot i if they are available during that slot, indicated by the availability parameter $a_{i,j} = 1$. Additionally, to ensure a balanced workload among teaching assistants, each TA is required to teach at least one tutorial session but no more than two tutorial sessions in total.

Let J denote the set of teaching assistants, indexed by j, and let I denote the set of available tutorial time slots, indexed by i. For each $i \in I$ and $j \in J$, we introduce two binary parameters: the availability indicator $a_{i,j}$, which equals 1 if TA j is available to teach during time slot i (and 0 otherwise), and the preference indicator $p_{i,j}$, which equals 1 if TA j prefers to teach in time slot i (and 0 otherwise).

Our objective is to determine a schedule that maximizes alignment with the preferences $p_{i,j}$ of the TAs while satisfying the scheduling constraints described above.

3 Methodology

We formulate the tutorial assignment problem as a mixed-integer linear program (MILP).

$$x_{i,j} = \begin{cases} 1, & \text{if TA } j \text{ is assigned to time slot } i, \\ 0, & \text{otherwise.} \end{cases}$$

The MILP is formulated as follows:

$$
\begin{aligned}
\text{maximize} \quad & \sum_{j\in\mathcal{J}}\sum_{i\in\mathcal{I}} p_{i,j}\, x_{i,j} && && \\
\text{subject to} \quad & \sum_{j\in\mathcal{J}} x_{i,j} = g_i, && \forall\, i \in \mathcal{I} && (1)\\
& x_{i,j} \le a_{i,j}, && \forall\, i \in \mathcal{I}, \forall\, j \in \mathcal{J} && (2)\\
& \sum_{i\in\mathcal{I}} x_{i,j} \le 2, && \forall\, j \in \mathcal{J} && (3)\\
& \sum_{i\in\mathcal{I}} x_{i,j} \ge 1, && \forall\, j \in \mathcal{J} && (4)\\
& x_{i,j} \in \{0,1\}, && \forall\, i \in \mathcal{I}, \forall\, j \in \mathcal{J} && (5)
\end{aligned}
$$

In this formulation, the objective is to maximize the total number of preference matches, where $p_{i,j}$ equals 1 if TA j prefers time slot i (and 0 otherwise). The first constraint guarantees that each time slot i receives exactly g_i TAs. The second constraint ensures that a TA can only be assigned to a time slot if available, as indicated by $a_{i,j}$. Finally, the third and fourth constraints control the workload by limiting each TA to at most two assignments while ensuring every TA is assigned to at least one tutorial.

4 Numerical Results

Our dataset consists of 21 TAs and 9 time slots. The model was solved on a laptop equipped with an Intel i9-13950HX using Python 3.9 and PuLP. With run times of 0.04 seconds for this instance, the results confirm that modern MIP solvers can efficiently handle moderately sized personnel scheduling problems (see also Bixby, 2012).

The solver achieved an *Optimal* status with maximal preference. It ensured feasibility by covering every time slot with the required number of TAs, attained a higher total preference with an objective value of 36 compared to 34 for the hand-crafted solution, and adhered strictly to capacity constraints by assigning each TA to at least one and no more than two tutorials.

To test the approach's robustness, we adjusted availability and TA preferences slightly. We introduced random perturbations to each data entry. Each variant reached optimal solutions, all under 0.06 seconds, indicating the method is flexible and adaptable to changing course demands, as highlighted in many moderately large personnel scheduling studies (e.g., Dowsland, 1998; Van den Bergh et al., 2013). Our results for the sensitivity analysis are given in the table below.

Instance	**Objective Value**	**Computational Time (s)**
1	35	0.05
2	35	0.01
3	Infeasible	0.03
4	31	0.02
5	34	0.05

Table 1: Sensitivity Analysis Results.

5 Conclusion

We have formulated a mixed-integer linear programming (MILP) approach to optimally assign teaching assistants across multiple tutorial time slots. Our model ensures that all sessions are adequately staffed, adheres strictly to capacity and availability constraints, and explicitly maximizes alignment with TA preferences. Numerical results demonstrate that this MILP solution significantly improves upon a manually generated schedule in terms of preference satisfaction without compromising feasibility. Moreover, sensitivity analysis confirms the robustness and adaptability of our model under varying input parameters. Future research could extend this framework by integrating additional considerations such as language proficiency or TA seniority and exploring heuristic or advanced optimization techniques to handle larger scheduling instances.

References

Bixby, R. E. (2012). A brief history of linear and mixed-integer programming computation. *Documenta Mathematica*, 2012:107–121.

Brucker, P., Qu, R., and Burke, E. (2011). Personnel scheduling: Models and complexity. *European Journal of Operational Research*, 210(3):467–473.

Dowsland, K. A. (1998). Nurse scheduling with tabu search and strategic oscillation. *European Journal of Operational Research*, 106(2-3):393–407.

Qu, X., Wang, S., Wang, T., Xiao, L., Yi, W., and Liu, Z. (2017). Mixed-integer linear programming models for teaching assistant assignment and extensions. *Scientific Programming*, 2017:1–10.

Van den Bergh, J., Beliën, J., De Bruecker, P., Demeulemeester, E., and De Boeck, L. (2013). Personnel scheduling: A literature review. *European Journal of Operational Research*, 226(3):367–385.