Mathematical Optimization Model for Weekly Task Scheduling

Formulation Documentation

April 8, 2025

1 Overview

This document presents the mathematical formulation of an optimization model designed to generate a weekly task schedule. The model assigns start times to tasks within a defined time horizon, respecting various constraints, and optimizes a weighted combination of maximizing leisure time and minimizing task-related stress.

Time Horizon

The planning horizon covers $D = 7$ days. Each day consists of $S_{day} = 56$ discrete time slots of 15 minutes each, typically representing the period from 8:00 AM to 10:00 PM. The total number of slots in the horizon is $S_{total} = D \times S_{day} = 392$. Slots are indexed globally from $s = 0$ to $S_{total} - 1$.

2 Parameters and Inputs

The model utilizes the following input parameters:

- **Set of Tasks (T):** The collection of tasks to be scheduled, indexed by i.
- **Task Attributes ($\forall i \in T$):**
  - p_i: Numerical priority of task i.
  - d_i: Numerical difficulty of task i.
  - dur_i: Duration of task i, measured in the number of time slots.
  - dl_i: Deadline for task i, represented as the global slot index by which the task must be fully completed.
  - $AllowedSlots_i$: A subset of $\{0, ..., S_{total} - 1\}$ indicating the permissible start slots for task i based on its time preference (e.g., "morning", "afternoon").
- **Set of Committed Slots (C):** A subset of $\{0, ..., S_{total} - 1\}$ representing time slots that are pre-allocated and unavailable for scheduling tasks (e.g., meetings, appointments).
- **Objective Weights:**
  - $\alpha \geq 0$: Weight coefficient emphasizing the maximization of leisure time.
  - $\beta \geq 0$: Weight coefficient emphasizing the minimization of the total stress score.
- **Daily Limit ($Limit_{daily}$, Optional):** An integer representing the maximum number of slots that can be occupied by tasks on any single day $d \in \{0, ..., D - 1\}$.

3 Decision Variables

The model determines the values of the following variables:

1. $X_{i,s}$ **(Binary):** Indicates if task i starts at slot s.

$$X_{i,s} = \begin{cases} 1 & \text{if task } i \in T \text{ starts at slot } s \in \{0, ..., S_{total} - 1\} \\ 0 & \text{otherwise} \end{cases}$$

2. Y_s **(Binary):** Indicates if slot s is occupied by any task.

$$Y_s = \begin{cases} 1 & \text{if slot } s \in \{0, ..., S_{total} - 1\} \text{ is occupied by a task} \\ 0 & \text{otherwise} \end{cases}$$

   This is an auxiliary variable linked to $X_{i,s}$ via constraints.

3. L_s **(Continuous):** Represents the leisure time (in minutes) available in slot s.

$$L_s \in [0, 15] \quad \forall s \in \{0, ..., S_{total} - 1\}$$

4 Objective Function

The objective is to maximize a weighted sum reflecting the trade-off between leisure and stress:

$$\text{Maximize} \quad Z = \alpha \sum_{s=0}^{S_{total}-1} L_s - \beta \sum_{i \in T} \sum_{s=0}^{S_{total}-1} (p_i \times d_i) X_{i,s}$$

Explanation:

- The first term, $\alpha \sum L_s$, promotes maximizing the total leisure time across all slots. The weight α scales the importance of leisure relative to stress reduction.
- The second term, $\beta \sum (p_i \times d_i) X_{i,s}$, aims to minimize the total "stress" incurred. Stress for each task i is defined as $p_i \times d_i$ and is counted if the task starts ($X_{i,s} = 1$). The weight β scales the penalty associated with scheduling high-priority or difficult tasks. This formulation associates stress with the initiation of a task, not its entire duration.

5 Constraints

The following constraints define the feasible schedules:

5.1 Task Assignment

Ensures that each task is assigned exactly one starting slot within the horizon.

$$\sum_{s=0}^{S_{total}-1} X_{i,s} = 1 \quad \forall i \in T$$

Explanation: For every task i, the sum of its binary start variables $X_{i,s}$ over all possible slots s must equal 1. This forces the model to select exactly one slot s where $X_{i,s} = 1$ (the start slot) for each task i.

5.2 Deadlines and Horizon

Ensures that tasks are completed by their deadline and fit entirely within the scheduling horizon.

$$X_{i,s} = 0 \quad \forall i \in T, \forall s \text{ such that } (s + dur_i - 1 > dl_i) \lor (s + dur_i > S_{total})$$

Explanation:

- **Deadline Check ($s+dur_i-1 > dl_i$):** A task i starting at slot s occupies slots from s to $s+dur_i-1$. The term $s + dur_i - 1$ represents the index of the last slot occupied by the task. This constraint forces the start variable $X_{i,s}$ to be 0 if the last occupied slot exceeds the task's specified deadline slot dl_i. This prevents tasks from starting if they cannot finish on time.
- **Horizon Check ($s+dur_i > S_{total}$):** The term $s+dur_i$ represents the index of the slot immediately following the task's completion. If this index is greater than the total number of slots S_{total}, it means the task would extend beyond the planning horizon. This constraint forces $X_{i,s}$ to 0 in such cases, ensuring tasks are fully contained within the schedule.

5.3 No Overlap

Prevents two or more tasks from being active in the same time slot.

$$\sum_{i \in T} \sum_{s=\max(0,t-dur_i+1)}^{t} X_{i,s} \leq 1 \quad \forall t \in \{0, ..., S_{total} - 1\}$$

Explanation: For any given time slot t, this constraint considers all tasks i. The inner sum iterates through potential start slots s for task i. The range $\max(0, t - dur_i + 1)$ to t identifies precisely those start slots s such that task i, if started at s, would be active during slot t. By summing the $X_{i,s}$ variables for all such task/start-slot combinations, we count how many tasks are active in slot t. Constraining this sum to be less than or equal to 1 ensures that at most one task can occupy any single slot t.

5.4 Preferences

Restricts the starting time of tasks to their allowed time windows (e.g., morning, afternoon).

$$X_{i,s} = 0 \quad \forall i \in T, \forall s \notin AllowedSlots_i$$

Explanation: For each task i, $AllowedSlots_i$ is the set of permissible start slots based on user preference. This constraint directly enforces the preference by setting the start variable $X_{i,s}$ to 0 for any slot s that is not within the task's allowed set. This prevents the task from starting outside its desired time window(s).

5.5 Commitments

Prevents any part of a scheduled task from coinciding with predefined fixed commitments.

$$X_{i,s} = 0 \quad \forall i \in T, \forall s \text{ such that } \{s, s + 1, ..., s + dur_i - 1\} \cap C \neq \emptyset$$

Explanation: For each task i and potential start slot s, the set $\{s, s+1, ..., s+dur_i-1\}$ represents all slots the task would occupy if it started at s. This constraint checks if this set has any overlap (non-empty intersection, $\cap C \neq \emptyset$) with the set of committed slots C. If an overlap exists, it means starting task i at s would conflict with a commitment. Therefore, the constraint forces $X_{i,s}$ to 0, preventing the task from starting at that time.

5.6 Leisure Calculation and Occupation Link (Y)

Links the auxiliary occupation variable Y_s to the task start variables $X_{i,s}$ and defines the leisure time L_s based on occupation and commitments.

$$
\begin{aligned}
&Y_s = \sum_{i \in T} \sum_{start=\max(0, s-dur_i+1)}^{s} X_{i,start} && \forall s \in \{0, ..., S_{total} - 1\} && (1)\\
&L_s = 0 && \forall s \in C && (2)\\
&L_s \leq 15 \times (1 - Y_s) && \forall s \notin C && (3)\\
&L_s \geq 0 && \forall s && (4)
\end{aligned}
$$

Explanation:

- **Equation (1):** This constraint defines the occupation variable Y_s. It equates Y_s to the same sum used in the "No Overlap" constraint. Since the No Overlap constraint ensures that this sum can only be 0 or 1, this equality correctly sets $Y_s = 1$ if slot s is occupied by any task, and $Y_s = 0$ otherwise. This makes Y_s a reliable binary indicator of task occupation for slot s.
- **Equation (2):** If slot s belongs to the set of commitments C, no leisure time is possible, so L_s is forced to 0.
- **Equation (3):** If slot s is not committed ($s \notin C$), this constraint limits the leisure time. If the slot is occupied by a task ($Y_s = 1$), then $1 - Y_s = 0$, forcing $L_s \leq 0$. Since L_s must also be non-negative (Eq. 4), L_s becomes exactly 0. If the slot is not occupied by a task ($Y_s = 0$), then $1 - Y_s = 1$, allowing $L_s \leq 15$. The objective function, seeking to maximize $\sum L_s$, will push L_s to its upper bound of 15 in this unoccupied, non-committed case.
- **Equation (4):** Explicitly states (or is implied by variable definition) that leisure time cannot be negative.

5.7 Daily Limits (Optional)

Enforces a maximum total duration of tasks scheduled within any single day.

Let $Slots_{day,d} = \{s \mid d \times S_{day} \leq s < (d+1) \times S_{day}\}$ be the set of slots for day d.

$$\sum_{s \in Slots_{day,d}} Y_s \leq Limit_{daily} \quad \forall d \in \{0, ..., D-1\}$$

Explanation: This constraint operates on a per-day basis ($d = 0$ to $D - 1$). For each day, it sums the occupation variables Y_s over all slots s belonging to that day ($Slots_{day,d}$). Since $Y_s = 1$ if a slot is occupied by a task and 0 otherwise, this sum represents the total number of task-occupied slots on day d. The constraint ensures this total does not exceed the specified $Limit_{daily}$.

6 Output

If the optimization problem is feasible and a solution is found (potentially optimal or feasible within a time limit), the model output provides the values of the decision variables. This typically translates to:

- The optimal objective function value Z.
- A schedule detailing the assigned start slot s (where $X_{i,s} = 1$) for each task i.
- The calculated total leisure time $\sum L_s$.
- The calculated total stress score $\sum (p_i \times d_i) X_{i,s}$.