Mathematical Optimization Model for Weekly Task Scheduling (Updated)

Formulation Documentation

April 12, 2025

1 Overview

This document presents the mathematical formulation of an optimization model designed to generate a weekly task schedule. The model first applies a pre-filter based on task duration, priority, and difficulty (the "Pi condition") to determine which tasks (T) are eligible for scheduling. It then assigns start times to all eligible 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 All Tasks (T_{all}):** The initial collection of tasks provided by the user, indexed by i.
- **Task Attributes ($\forall i \in T_{all}$):**
  - p_i: Numerical priority of task i.
  - d_i: Numerical difficulty of task i. Assumed $d_i > 0, p_i > 0$.
  - dur_slots_i: Duration of task i, measured in the number of time slots (1 slot = 15 minutes).
  - dur_min_i: Duration of task i, measured in minutes ($dur\_min_i = dur\_slots_i \times 15$).
  - 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").
  - $hard\_threshold$: Integer difficulty rating at or above which a task is considered "hard" (default: 4).
- **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 Task Pre-filtering (Pi Condition)

Before optimization, tasks are filtered based on a derived "Probability of Importance" (Pi) condition. Only tasks meeting this condition are considered schedulable.

Definition 1 (Schedulable Tasks (T))**.** *A task $i \in T_{all}$ is considered schedulable and included in the set T (where $T \subseteq T_{all}$) if it satisfies the following condition, which is derived from the original requirement that the probability of completion (P_i) must be at least 0.7:*

$$dur\_min_i \geq (d_i \times p_i) \times \ln\left(\frac{10}{3}\right) \tag{1}$$

Tasks in T_{all} that do not meet this condition are excluded from the optimization process. Let $T = \{i \in T_{all} \mid$ Eq. (1) holds$\}$.

Derivation of the Pi Condition: The condition stems from an underlying model assumption (not explicitly part of the MILP formulation, but used as a pre-filter) that a task i is only viable if its estimated probability of completion (P_i) within the allocated time (dur_min_i) is sufficient. This probability was defined in earlier documentation as:

$$P_i = 1 - \exp\left(-\lambda_i \times \frac{dur\_min_i}{d_i \times p_i}\right)$$

where λ_i is a rate parameter (assumed to be $\lambda_i = 1$), d_i is difficulty, and p_i is priority (urgency). The required threshold is $P_i \geq 0.7$.

Setting $\lambda_i = 1$ and applying the threshold:

$$1 - \exp\left(-\frac{dur\_min_i}{d_i \times p_i}\right) \geq 0.7$$

$$0.3 \geq \exp\left(-\frac{dur\_min_i}{d_i \times p_i}\right)$$

Taking the natural logarithm of both sides (which preserves the inequality direction):

$$\ln(0.3) \geq -\frac{dur\_min_i}{d_i \times p_i}$$

Multiply by -1 and reverse the inequality sign:

$$-\ln(0.3) \leq \frac{dur\_min_i}{d_i \times p_i}$$

Using the property $-\ln(x) = \ln(1/x)$:

$$\ln\left(\frac{1}{0.3}\right) \leq \frac{dur\_min_i}{d_i \times p_i}$$

$$\ln\left(\frac{10}{3}\right) \leq \frac{dur\_min_i}{d_i \times p_i}$$

Rearranging for dur_min_i (assuming $d_i > 0, p_i > 0$):

$$dur\_min_i \geq (d_i \times p_i) \times \ln\left(\frac{10}{3}\right)$$

This yields the inequality used in Definition 1.

Interpretation: This filter removes tasks that are deemed too short relative to their combined difficulty and priority to meet the 70% completion probability threshold under the assumed model. The constant $\ln(10/3) \approx 1.204$. Tasks that fail this check are reported as filtered out. The subsequent model formulation operates only on the set of schedulable tasks T.

4 Decision Variables

The model determines the values of the following variables for the set of **schedulable tasks** T:

1. $X_{i,s}$ **(Binary):** Indicates if schedulable task $i \in T$ 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 schedulable task $i \in T$.

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

   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.

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

5 Objective Function

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

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

Explanation:

- The first term, $\alpha \sum L_s$, promotes maximizing the total leisure time across all slots.
- The second term, $\beta \sum (p_i \times d_i) X_{i,s}$, represents the total "stress" incurred from the scheduled tasks in set T. Since Constraint 2 requires all tasks $i \in T$ to be scheduled (i.e., $\sum_s X_{i,s} = 1$ for each $i \in T$), the total stress sum $\sum_{i \in T} (p_i \times d_i)$ becomes a constant value once the set T is determined by the pre-filter. Therefore, minimizing this term primarily serves to break ties between solutions that achieve the same maximum leisure, potentially favoring schedules where stress (if variable per task instance, which it isn't here) is lower, or simply acts as a constant offset if $\beta > 0$. The primary optimization driver is typically the leisure term.

6 Constraints

The following constraints define the feasible schedules for the set of **schedulable tasks** T:

6.1 Mandatory Task Assignment

Ensures every task that passed the pre-filter (i.e., is in set T) is scheduled exactly once.

$$\sum_{s=0}^{S_{total}-1} X_{i,s} = 1 \quad \forall i \in T \tag{2}$$

Explanation: For every task i in the set of schedulable tasks T, exactly one start slot s must be chosen ($X_{i,s} = 1$). This enforces that all tasks deemed viable by the pre-filter are included in the final schedule. If this constraint cannot be met simultaneously with other constraints (e.g., due to lack of available slots), the model will be infeasible.

6.2 Hard Task Limitation

Restricts the number of difficult tasks (from set T) starting on any single day.

$$\sum_{i \in T_{\text{hard}}} \sum_{s \in \text{Slots}_{day,d}} X_{i,s} \leq 1 \quad \forall d \in \{0, ..., D-1\}$$

where $T_{\text{hard}} = \{i \in T : d_i \geq \text{hard\_threshold}\}$ is the subset of *schedulable* tasks with difficulty ratings at or above the threshold. $Slots_{day,d}$ is the set of global slot indices belonging to day d.

Explanation: For each day d, this sums the start variables $X_{i,s}$ only for hard tasks within the schedulable set T that start on day d. Limiting this sum to at most 1 ensures no more than one hard (and schedulable) task begins on any single day.

6.3 Deadlines and Horizon

Ensures that scheduled tasks (from set T) are completed by their deadline and fit entirely within the scheduling horizon.

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

Explanation: This prevents a task $i \in T$ from starting at slot s if its last slot $(s + dur\_slots_i - 1)$ would occur after its deadline slot dl_i, or if the task duration causes it to extend beyond the total number of slots S_{total}. Note that $s + dur\_slots_i > S_{total}$ is equivalent to $s + dur\_slots_i - 1 \geq S_{total}$, meaning the last slot is outside the valid range $[0, S_{total} - 1]$.

6.4 No Overlap

Prevents two or more scheduled tasks (from set T) from being active in the same time slot.

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

Explanation: For any time slot t, this sums the start variables $X_{i,start}$ for all tasks $i \in T$ that would be active during slot t. A task i started at $start$ is active during slot t if $start \leq t < start + dur\_slots_i$. The inner summation correctly identifies these relevant start times. Constraining the sum to be ≤ 1 ensures mutual exclusivity.

6.5 Preferences

Restricts the starting time of scheduled tasks (from set T) to their allowed time windows.

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

Explanation: Enforces the start variable $X_{i,s}$ to be 0 for any task $i \in T$ if slot s is outside its permitted preference set $AllowedSlots_i$.

6.6 Commitments

Prevents any part of a scheduled task (from set T) from coinciding with predefined fixed commitments (set C).

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

Explanation: Prevents task $i \in T$ from starting at s if any slot it would occupy during its duration (from s to $s + dur\_slots_i - 1$) overlaps with the set of committed slots C.

6.7 Leisure Calculation and Occupation Link (Y)

Links the auxiliary occupation variable Y_s to the task start variables $X_{i,s}$ for schedulable tasks (from set T) and defines leisure L_s.

$$
\begin{aligned}
&Y_s = \sum_{i \in T} \sum_{start=\max(0,s-dur\_slots_i+1)}^{s} X_{i,start} && \forall s \in \{0, ..., S_{total} - 1\} && (3)\\
&L_s = 0 && \forall s \in C && (4)\\
&L_s \leq 15 \times (1 - Y_s) && \forall s \notin C && (5)\\
&L_s \geq 0 && \forall s && (6)
\end{aligned}
$$

Explanation:

- Equation (3) defines Y_s to be 1 if slot s is occupied by any task $i \in T$, and 0 otherwise. This re-uses the logic from the No Overlap constraint (6.4) to determine occupation at slot s. Because of constraint 6.4, the sum on the right-hand side can only be 0 or 1, making Y_s effectively binary.
- Equation (4) sets leisure to 0 for committed slots.
- Equation (5) allows leisure up to 15 minutes only if the slot s is not committed (checked implicitly by applying this only to $s \notin C$) AND not occupied by a task ($Y_s = 0$). The objective function seeks to maximize L_s, so it will push L_s to 15 if allowed by this constraint.
- Equation (6) ensures leisure is non-negative (though $L_s \in [0, 15]$ is defined earlier).

6.8 Daily Limits (Optional)

Enforces a maximum total number of task-occupied slots (by tasks from T) within any single day.

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

where $Slots_{day,d}$ is the set of global slot indices belonging to day d.

Explanation: For each day d, the total number of slots s occupied by tasks from T (indicated by $Y_s = 1$) must not exceed the optional parameter $Limit_{daily}$.

7 Output

If the optimization problem is feasible and a solution is found, the model output provides:

- The optimal objective function value Z.
- A schedule detailing the assigned start slot s (where $X_{i,s} = 1$) for each schedulable task $i \in T$.
- The calculated total leisure time $\sum L_s$.
- The calculated total stress score based on scheduled tasks: $\sum_{i \in T}(p_i \times d_i)$. (Note: This is constant for a given set T).
- A list of tasks from the original set T_{all} that were filtered out by the Pi condition and thus not considered for scheduling (set $T_{all} \setminus T$). Includes the reason for filtering.
- The effective completion rate, calculated as $|T|/|T_{all}|$, representing the fraction of initially provided tasks that were deemed schedulable by the filter.

If no feasible schedule exists for the set of tasks T (meaning it's impossible to schedule all tasks in T while respecting all constraints), the model reports infeasibility.