

Mathematical Optimization Model for Weekly Task Scheduling with Deadline Penalties

Formulation Documentation

April 13, 2025

1 Overview

This document presents the mathematical formulation of an optimization model designed to generate a weekly task schedule with deadline penalties. 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, with an additional penalty for scheduling tasks close to their deadlines. This model extends the base model by adding a deadline proximity factor that increases the effective stress of tasks scheduled near their deadlines.

Time Horizon

The planning horizon covers $D = 7$ days. Each day consists of S_{day} discrete time slots of 15 minutes each, typically representing the period from 8:00 AM to 10:00 PM (configurable). The total number of slots in the horizon is $S_{total} = D \times S_{day}$. 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 $\{0, \dots, S_{total} - 1\}$ 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 $\{0, \dots, S_{total} - 1\}$ 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.
 - $\gamma \geq 0$: Weight coefficient controlling the impact of deadline proximity on stress (new in this model).
- **Daily Limit** (*Limit_{daily}*, **Optional**): An integer representing the maximum number of slots that can be occupied by tasks on any single day $d \in \{0, \dots, D - 1\}$.

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:*

$$dur_min_i \geq (d_i \times p_i) \times \ln\left(\frac{10}{3}\right) \quad (1)$$

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

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.

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 .

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

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

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

5 Deadline Penalty Factor

This model introduces a deadline penalty factor that increases the stress associated with scheduling tasks closer to their deadlines. This factor is calculated for each task $i \in T$ and potential start slot s .

Definition 2 (Deadline Penalty Factor). *For a task $i \in T$ with deadline slot dl_i and duration dur_slots_i , the deadline penalty factor $\phi_{i,s}$ for starting at slot s is defined as:*

$$\phi_{i,s} = \begin{cases} \frac{s}{LPS_i} & \text{if } LPS_i > 0 \\ 0 & \text{if } LPS_i = 0 \end{cases} \quad (2)$$

where $LPS_i = \max(0, dl_i - dur_slots_i + 1)$ is the latest possible start slot that allows task i to complete by its deadline.

Properties of the Deadline Penalty Factor:

- $\phi_{i,s} \in [0, 1]$ for all valid start slots s (ensured by the deadline constraint).
- $\phi_{i,s} = 0$ when $s = 0$ (starting at the earliest possible slot).
- $\phi_{i,s} = 1$ when $s = LPS_i$ (starting at the latest possible slot before the deadline).
- If $LPS_i = 0$ (i.e., the task can only start at slot 0 to meet its deadline), then $\phi_{i,s} = 0$.

6 Objective Function with Deadline Penalty

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

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

Explanation:

- The first term, $\alpha \sum L_s$, promotes maximizing the total leisure time across all slots.
- The second term (the stress term) now includes the deadline penalty factor:

$$\beta \sum_{i \in T} \sum_{s=0}^{S_{total}-1} (p_i \times d_i) \times (1 + \gamma \times \phi_{i,s}) \times X_{i,s}$$

- The base stress for task i is still $p_i \times d_i$, but now multiplied by a factor $(1 + \gamma \times \phi_{i,s})$ which increases as the task is scheduled closer to its deadline.

- $\gamma \geq 0$ is a parameter that controls the strength of the deadline penalty. Larger values of γ place more emphasis on scheduling tasks earlier rather than later.
- When $\gamma = 0$, this model reduces to the base model without deadline penalties.
- When $\gamma > 0$, the effective stress of scheduling task i at slot s can increase by up to a factor of $(1 + \gamma)$ when the task is scheduled at its latest possible start time.

Example: For a task with priority 3 and difficulty 2 (base stress = 6), with $\gamma = 1$:

- If scheduled at slot 0 (earliest): $(1 + 1 \times 0) \times 6 = 6$ (no penalty)
- If scheduled halfway to deadline ($\phi_{i,s} = 0.5$): $(1 + 1 \times 0.5) \times 6 = 9$ (50% increase)
- If scheduled at latest possible slot ($\phi_{i,s} = 1.0$): $(1 + 1 \times 1.0) \times 6 = 12$ (100% increase)

7 Constraints

The constraints in the deadline penalty model remain the same as in the base model. They define the feasible schedules for the set of **schedulable tasks** T :

7.1 Mandatory Task Assignment

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

$$\sum_{s=0}^{S_{total}-1} X_{i,s} = 1 \quad \forall i \in T \quad (3)$$

7.2 Hard Task Limitation

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

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

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. $\text{Slots}_{\text{day},d}$ is the set of global slot indices belonging to day d .

7.3 Deadlines and Horizon

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

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

Note: This constraint is particularly important for the deadline penalty model, as it defines the feasible range of start slots for which the deadline penalty factor $\phi_{i,s}$ is calculated.

7.4 No Overlap

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

$$\sum_{i \in T} \sum_{\text{start}=\max(0, t-\text{dur_slots}_i+1)}^t X_{i,\text{start}} \leq 1 \quad \forall t \in \{0, \dots, S_{total}-1\} \quad (4)$$

7.5 Preferences

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

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

7.6 Commitments

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

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

7.7 Leisure Calculation

Defines leisure L_s based on commitments and task occupation.

$$L_s = 0 \quad \forall s \in C \quad (5)$$

$$L_s \leq 15 \times \left(1 - \sum_{i \in T} \sum_{start=\max(0, s-dur_slots_i+1)}^s X_{i,start} \right) \quad \forall s \notin C \quad (6)$$

$$L_s \geq 0 \quad \forall s \quad (7)$$

7.8 Daily Limits (Optional)

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

$$\sum_{i \in T} \sum_{start=0}^{S_{total}-1} (X_{i,start} \times SlotsInDay(start, dur_slots_i, d)) \leq Limit_{daily} \quad \forall d \in \{0, \dots, D-1\}$$

where $Slots_{day,d}$ is the set of global slot indices belonging to day d (from $d \times S_{day}$ to $(d+1) \times S_{day}-1$), and $SlotsInDay(start, duration, day)$ calculates the number of slots occupied by a task starting at $start$ with $duration$ that fall within day d .

8 Implementation Considerations

8.1 Dynamic Scheduling Window

The model supports a configurable daily scheduling window by adjusting S_{day} based on the desired start and end hours. This flexibility allows users to define different active periods (e.g., 8am-10pm, 9am-5pm) while maintaining the same mathematical formulation.

8.2 Gamma Parameter Selection

The γ parameter controls the trade-off between scheduling tasks early versus maximizing leisure or minimizing base stress:

- $\gamma = 0$: No penalty for late scheduling; equivalent to the base model
- $\gamma \approx 1$: Balanced approach; tasks scheduled at their latest possible time have their stress doubled
- $\gamma > 1$: Strong preference for early scheduling; dramatically increases stress for tasks scheduled close to deadlines

8.3 Computational Impact

The deadline penalty factor doesn't introduce additional constraints or variables compared to the base model, but it does make the objective function more complex. This generally has minimal impact on solving times, as it only affects the coefficients in the existing stress term.

9 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, including the deadline penalty:

$$\sum_{i \in T} (p_i \times d_i \times (1 + \gamma \times \phi_{i,s_i}))$$

where s_i is the assigned start slot for task i .

- 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.

10 Benefit Analysis

The deadline penalty model offers several advantages over the base model:

- **Proactive Time Management:** By penalizing scheduling close to deadlines, the model encourages completing tasks well before they're due, reducing last-minute pressure.
- **Priority-Weighted Urgency:** Higher priority/difficulty tasks face stronger deadline penalties, causing the model to schedule these important tasks earlier than less important ones, even with similar deadlines.
- **Procrastination Mitigation:** The model systematically discourages pushing tasks to their deadlines, mimicking good time management practices.
- **Customizable Urgency:** The γ parameter provides a tunable control over how strongly the model avoids scheduling tasks close to their deadlines.
- **Forward-Looking Schedules:** The model produces more balanced schedules that distribute workload more evenly across the time horizon, reducing stress peaks.

The deadline penalty model maintains all the functionality of the base model while adding a natural bias toward earlier task completion. This matches the intuition that even when a task technically "can wait," it's often better to handle it sooner rather than later.