Time Lords

OptiTime: Intelligent Campus Timetable Optimization

BT2024243 — Shivansh Shah — BT2024239 — Ravi Abhinav — BT2024221 — R.V. Pranav

Semester Project — Optimization Techniques

Problem statement

OptiTime mathematically models scheduling as an MILP to produce clash-free timetables that reduce student fatigue, balance faculty load, and improve room use.

Planned method (model & approach)

We represent every possible assignment as a binary variable:

$$x_{c,t,r} \in \{0,1\} \quad = 1 \text{ if course } c \text{ is scheduled at timeslot } t \text{ in room } r.$$

Course slot requirement. Each course c requires h_c slots (input from credits and slot length). We enforce:

$$\sum_{t \in T} \sum_{r \in R} x_{c,t,r} = h_c \quad \forall c.$$

Core linear constraints:

$$\sum_{c} x_{c,t,r} \leq 1 \quad \forall t, r \quad \text{(no room double-booking)}$$

$$\sum_{c \in C_p} \sum_{r} x_{c,t,r} \leq 1 \quad \forall p, t \quad \text{(professor no-overlap)}$$

If courses c, c' share students:

$$\sum_{r} x_{c,t,r} + \sum_{r} x_{c',t,r} \leq 1 \quad \forall t.$$

For room fit: set $x_{c,t,r} = 0$ if $\text{cap}_r < e_c$.

Contiguous multi-hour sessions (optional): For a course needing a single block of length k, introduce start variables $s_{c,t,r} \in \{0,1\}$ for $t = 1, \ldots, |T| - k + 1$:

$$\sum_{t} \sum_{r} s_{c,t,r} = 1, \qquad x_{c,t+i,r} \geq s_{c,t,r} \ (i = 0, \ldots, k-1),$$

and $\sum_{t,r} x_{c,t,r} = k$. This forces k consecutive slots.

Same-room option (optional): To require a single room for all meetings of c, add $y_{c,r} \in \{0,1\}$ with

$$x_{c,t,r} \leq y_{c,r} \quad \forall t, \qquad \sum_{r} y_{c,r} = 1.$$

Objective (linear, tunable). Minimize a weighted sum:

$$\min \ W_1 \sum_{s,t} z_{s,t} + W_2 \sum_{p} (u_p^+ + u_p^-) + W_3(\text{unused seats}) + W_4(\text{pref penalty})$$

where $z_{s,t}$ marks back-to-back classes for student cohort s (to cut fatigue), $u_p^{\pm}$ linearize professor-load deviation, and pref penalty encodes soft professor time preferences.

Data, tools & outputs

Inputs: course list (with h_c), enrollments (or conflict matrix), room list (capacity/type), professor-course mapping and availability/preferences, timeslot grid.
Tools: Python + Pandas (preprocessing), PuLP/Pyomo or OR-Tools (modeling), CBC baseline solver (Gurobi if available), Matplotlib/Plotly or Streamlit for demo.
Outputs: course → (timeslot, room) assignments, per-student and per-professor timetables, and evaluation metrics.