5 Multi-phase Approach

We solve the course timetabling problem in a sequence of four separate phases. Phase-1 assigns instruction-units to instructors. phase-2 assigns lectures to day-sequences, phase-3 assigns labs and tutorials to time-slots and finally phase-4 assigns labs and tutorials to days and available time-slots in the days.

Each phase is solved using constraint programming with suitable heuristics for ordering the decision variables, and maximizes a subjective preference function over a given set of constraints and preferences. In the objective function of each phase we use a weight function, which gives higher weight when the concerned preference is satisfied.

5.1 Phase-1: Assigning Courses to Instructors

Phase-1 has two sub-phases. Phase-1a assigns lectures to professors. Phase-1b assigns labs and tutorials to academic assistants and graduate students (teaching assistants).

In the mathematical formulation of phase-1 problem presented below, the decision variable \( X_{c,d} \) is the indicator of course \( c \) assigned to day-sequence \( d \), and \( 0 \) otherwise. \( W_c \) is the weight of course \( c \) on instructor \( i \) on course time.

\[
\text{maximize} \sum_{c \in C} \sum_{d \in Days} X_{c,d} W_c \tag{1}
\]

subject to,
\[
\sum_{c \in C} X_{c,d} = \text{Limit}_i, \quad i \in I \tag{2}
\]

Here, \( \text{Limit}_i \) is the maximum number of course that instructor \( i \) may teach. For phase-1a, \( C \) is the set of lectures and \( I \) is the set of professors. For phase-1b, \( I \) is the set of labs and tutorials and \( 1 \) is the set of academic assistants and graduate students.

The objective function (Equation 1) satisfies the instructors' preferences on courses as much as possible.

Equation 2 imposes that the number of instruction-unit assigned to any instructor must not exceed the maximum capacity of the instructor.

5.2 Phase-2: Assigning Lectures to Day-sequences

In phase-2 lectures are assigned to one of the two day-sequences while satisfying the instructors' preferences on day-sequences as much as possible. The mathematical formulation is given below. The decision variable \( X_{c,d} \) if lecture \( c \) is assigned to day-sequence \( d \) and \( 0 \) otherwise.

\[
\text{maximize} \sum_{c \in C} \sum_{d \in Days} X_{c,d} W_c \tag{3}
\]

subject to,
\[
\sum_{c \in C} X_{c,d} \leq \text{Slots}_d, \quad d \in Days \tag{4}
\]

Here, \( \text{Slots}_d \) is the set of courses that the day-sequence \( d \) and \( \text{Res}_{c,d} \) are respectively the number of time-slots and courses rooms available in a day of day-sequence \( d \).

The objective function (Equation 4) satisfies professors' preferences on day-sequences as much as possible.

Equation 5 restricts that the number of lecture \( c \) assigned to a day-sequence does not exceed the number of available time-slots in that day-sequence.

Equation 6 ensures that each lecture is assigned to exactly one day-sequence.

5.3 Phase-3: Assigning Lectures to Time-slots

After the assignment of lectures to day-sequences in phase-2 is complete, phase-3 operates on lectures of a single day-sequence at a time and assigns them to available time-slots. The mathematical formulation of phase-3 problem is given below. The decision variable \( X_{c,d,t} \) if lecture \( c \) is assigned to time-slot \( t \) and \( 0 \) otherwise.

\[
\text{maximize} \sum_{c \in C} \sum_{d \in Days} \sum_{t \in TimeSlots} X_{c,d,t} \tag{5}
\]

subject to,
\[
\sum_{t \in TimeSlots} X_{c,d,t} = 1, \quad c \in Lecture \tag{6}
\]

Here, \( \text{Lecture} \) is the set of lectures assigned to day-sequence \( d \) and \( \text{Slots}_{c,d,t} \) is the set of time-slots available in any day of day-sequence \( d \); \( \text{Res}_{c,d,t} \) is the number of time-slots and course rooms available in that day-sequence.

The objective function (Equation 7) satisfies professors' preferences on time as much as possible.

Equation 8 ensures that each lecture is assigned to exactly one time-slot.

Equation 9 restricts that the number of lectures assigned to any time-slot does not exceed the number of available classrooms during that period.

Equation 10 enforces that the lectures of the same section are not scheduled in over-lapping time-slots.

Equation 11 imposes at least one time-slot gap between lectures of the same instructor.

5.4 Phase-4: Labs and Tutorials to Days and Times

Phase-4 is the last phase. In this phase, labs and tutorials are assigned to one of the five week-days and to available-time-slots within the days.

In the mathematical formulation of phase-4 stated below, the decision variable \( X_{c,d,t} \) if instruction-unit \( c \) is assigned to time-slot \( t \) of day \( d \) and \( 0 \) otherwise.

\[
\text{maximize} \sum_{c \in Labs \cup Tutorials} \sum_{d \in Days} \sum_{t \in TimeSlots} X_{c,d,t} W_c \tag{12}
\]

subject to,
\[
\sum_{d \in Days} \sum_{t \in TimeSlots} X_{c,d,t} = \text{Limit}_c, \quad c \in Labs \cup Tutorials \tag{13}
\]

Here, \( \text{Limit}_c \) is the number of instruction-unit assigned to any instruction-unit. \( c \) is the set of labs and tutorials and \( 1 \) is the set of academic assistants and graduate students.

The objective function (Equation 12) satisfies the instructors' preferences on time as much as possible.

Equation 13 ensures that each lab or tutorial is assigned to the number of time-slots of each instruction-unit belonging to a single section.

Equation 14 imposes that each lab or tutorial is assigned to time-slots of the same day.

Equation 15 restricts each lab or tutorial assigned to consecutive time-slots.

Equation 16 ensures that the number of labs and tutorials assigned to any time-slot does not exceed the number of classrooms available during that period.

Equation 17 imposes that there should be a gap of at least one time-slot between courses taught by the same instructor.

Equation 18 enforces that the labs and tutorials of the section are not scheduled in overlapping time-slots.

Equation 19 ensures that labs and tutorials are scheduled in time-slots not over-lapping with the lectures of the same section.

6 Implementation

The model of each phase is implemented in OPL (Optimization Programming Language). A commercial solver ILOG’s CPLEX is used for solving the subproblem in each phase. A C++ module is used to integrate different phases and enable data flow between phases. Figure 1 shows the architecture of our timetabling implementation.

![Figure 1: Architecture of the timetabling implementation](image-url)