DM872

Math Optimization at Work

Course Timetabling

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

Outline

Outline

1. Introduction
2. School Timetabling
3. Course Timetabling
    - Formalization and Modelling

Educational timetabling process

Phase:	Planning	Scheduling	Dispatching
Horizon:	Long Term	Timetable Period	Day of Operation
Objective:	Service Level	Feasibility, Quality	Get it Done
Steps:	Manpower, Curriculum, Facilities, Equipment	Quarterly Timetabling, Project assignment, student sectioning	Repair

Timetabling

Assignment of events to a limited number of time periods and locations subject to constraints

Two categories of constraints:

Hard constraints $H = \{H_1, \ldots, H_n\}$: must be strictly satisfied, no violation is allowed

Soft constraints $\Sigma = \{S_1, \ldots, S_m\}$: their violation should be minimized (determine quality)

Each institution may have some unique combination of hard constraints and take different views on what constitute the quality of a timetable.

Outline

School Timetabling

[aka, teacher-class model]

The daily or weekly scheduling for all the classes of a high school, avoiding teachers meeting two classes in the same time.

Input:

- a set of classes $\mathcal{C} = \{C_1, \ldots, C_m\}$
  A class is a set of students who follow exactly the same program. Each class has a dedicated room.
- a set of teachers $\mathcal{P} = \{P_1, \ldots, P_n\}$
- a requirement matrix $\mathcal{R}_{m \times n}$ where R_{ij} is the number of lectures given by teacher P_j to class C_i.
- all lectures have the same duration (say one period)
- a set of time slots $\mathcal{T} = \{T_1, \ldots, T_p\}$ (the available periods in a day).

Output: An assignment of lectures to time slots such that no teacher or class is involved in more than one lecture at a time

A Simple Model (daily or weekly)

IP formulation:

Binary variables: assignment of class C_i to teacher P_j in T_k

$$x_{ijk} = \{0,1\} \ \forall i = 1,\dots,m; \ j = 1,\dots,n; \ k = 1,\dots,p$$

Constraints:

$$\sum_{k=1}^{p} x_{ijk} = R_{ij} \ \forall i = 1,\dots,m; \ j = 1,\dots,n$$

$$\sum_{j=1}^{n} x_{ijk} \leq 1 \quad \forall i = 1,\dots,m; \ k = 1,\dots,p$$ class: at most one per teacher per slot

$$\sum_{i=1}^{m} x_{ijk} \leq 1 \quad \forall j = 1,\dots,n; \ k = 1,\dots,p$$ teacher: at most one class per slot

Graph model

Bipartite multigraph $G = (\mathcal{C}, \mathcal{P}, \mathcal{R})$:

- nodes $\mathcal{C}$ and $\mathcal{P}$: classes and teachers
- R_{ij} parallel edges

Time slots are colors ➜ Graph-Edge Coloring problem

Theorem: [König] There exists a solution that uses p colors iff:

$$\sum_{i=1}^{m} R_{ij} \leq p \; \forall j = 1, \ldots, n$$
$$\sum_{i=1}^{n} R_{ij} \leq p \; \forall i = 1, \ldots, m$$

Weekly Model

The p timeslots are the days and R_{ij} the weekly requirements

- a_i max number of lectures for a class in a day
- b_j max number of lectures for a teacher in a day

IP formulation:

Variables: number of lectures to a class in a day

$$x_{ijk} \in N \; \forall i = 1, \ldots, m; \; j = 1, \ldots, n; \; k = 1, \ldots, p$$

Constraints:

$$\sum_{k=1}^{p} x_{ijk} = R_{ij} \; \forall i = 1, \ldots, m; \; j = 1, \ldots, n$$

$$\sum_{i=1}^{m} x_{ijk} \leq b_j \;\; \forall j = 1, \ldots, n; \; k = 1, \ldots, p$$

$$\sum_{j=1}^{n} x_{ijk} \leq a_i \;\; \forall i = 1, \ldots, m; \; k = 1, \ldots, p$$

Graph model

Edge coloring model still valid but with

- no more than a_i edges adjacent to C_i have same colors and
- and more than b_j edges adjacent to T_j have same colors

Theorem: [König] There exists a solution that uses p slots iff:

$$\sum_{i=1}^{m} R_{ij} \leq b_j p \ \forall j = 1, \ldots, n$$
$$\sum_{j=1}^{n} R_{ij} \leq a_i p \ \forall i = 1, \ldots, m$$

1. find minimum number of periods needed $p = \max\left\{\max_j \left\lceil \sum_{i=1}^{m} R_{ij}/b_j \right\rceil, \max_i \left\lceil \sum_{j=1}^{n} R_{ij}/a_i \right\rceil \right\}$
2. if p is given, find a formulation of the problem that admits a solution balancing the work load (see next slide)

Balanced Weekly Model

Given the p days and R_{ij} the weekly requirements

IP formulation:

Variables: number of lectures to a class in a day

$$
\begin{array}{rl}
x_{ijk} \in N & \forall i = 1..m;\ j = 1..n;\ k = 1..p \\
\lfloor R_{ij}/p \rfloor \leq x_{ijk} \leq \lceil R_{ij}/p \rceil & \forall i = 1..m;\ j = 1..n;\ k = 1..p
\end{array}
$$

Constraints:

$$
\begin{array}{ll}
\sum_{k=1}^{p} x_{ijk} = R_{ij} & \forall i = 1, \ldots, m;\ j = 1, \ldots, n \\
\left\lfloor \sum_{i=1}^{m} R_{ij}/p \right\rfloor \leq \sum_{i=1}^{m} x_{ijk} \leq \left\lceil \sum_{i=1}^{m} R_{ij}/p \right\rceil & \forall j = 1, \ldots, n;\ k = 1, \ldots, p \\
\left\lfloor \sum_{j=1}^{n} R_{ij}/p \right\rfloor \leq \sum_{j=1}^{n} x_{ijk} \leq \left\lceil \sum_{j=1}^{n} R_{ij}/p \right\rceil & \forall i = 1, \ldots, m;\ k = 1, \ldots, p
\end{array}
$$

$\rightsquigarrow$ The edge coloring problem in the multigraph is solvable in polynomial time by solving a sequence of p network flows problems. [De Werra, 1985]

Possible approach: solve the weekly timetable first and then the daily timetable

Further constraints that may arise:

- Preassignments
- Unavailabilities
  (can be expressed as preassignments with dummy class or teachers)

They make the problem NP-complete if any teacher is unavailable during more than 2 periods.
(Reduction from 3-SAT, [Even, Itai, Shamir, 1975])

- Bipartite matchings can still help in developing heuristics, for example, for solving x_{ijk} keeping any index fixed.

Further complications:

- Simultaneous lectures (eg, gymnastic)
- Subject issues (more teachers for a subject and more subjects for a teacher)
- Room issues (use of special rooms)

So far feasibility problem.

Preferences (soft constraints) may be introduced

- Desirability of assigning teacher P_j to class C_i in T_k

$$\min \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{p} d_{ijk} x_{ijk}$$

- Organizational costs: having a teacher available for possible temporary teaching posts
- Specific day off for a teacher

Optimization criteria

Introducing soft constraints the problem becomes a multiobjective problem.

Possible ways of dealing with multiple objectives:

- weighted sum
- lexicographic order
- distance from optimal or nadir point
- Pareto-frontier

Optimization criteria

Taking into account individual preferences, the problem can be seen as a collective welfare problem:

- Pareto efficiency (gives only partial ordering)
- classical utilitarian ordering (weak ordering)
  weighted sum
- egalitarian ordering (weak ordering)
  leximin ordering, generous maximum allocations (or maximum cardinality, rank maximal matching) and greedy maximum allocations
- fair allocation:
  - just allocation process (procedural justice
  - just outcome of the allocation (distributive justice)
    individual welfare: minimax (minimize maximal discontent)
- combinations thereof

Heuristic Methods

Construction heuristic

Based on principles:

- most-constrained lecture on first (earliest) feasible timeslot
- most-constrained lecture on least constraining timeslot

Enhancements:

- limited backtracking
- local search optimization step after each assignment

More later

Local Search Methods and Metaheuristics

High level strategy:

- Single stage (hard and soft constraints minimized simultaneously)
- Two stages (feasibility first and quality second)

Dealing with feasibility issue:

- partial assignment: do not permit violations of H but allow some lectures to remain unscheduled
- complete assignment: schedule all the lectures and seek to minimize H violations

Outline

Outline

Course Timetabling

The weekly scheduling of the lectures/events/classes of courses avoiding students, teachers and room conflicts.

Input:

- A set of courses $\mathcal{C} = \{C_1, \ldots, C_n\}$ each consisting of a set of lectures $\mathcal{C}_i = \{L_{i1}, \ldots, L_{il_i}\}$. Alternatively,
  A set of lectures (aka, events, classes) $\mathcal{L} = \{L_1, \ldots, L_l\}$.
- A set of curricula $\mathcal{S} = \{S_1, \ldots, S_r\}$ that are groups of courses with common students (curriculum based model). Alternatively,
  A set of enrollments $\mathcal{S} = \{S_1, \ldots, S_s\}$ that are groups of courses that a student wants to attend (Post enrollment model).
- a set of time slots $\mathcal{T} = \{T_1, \ldots, T_p\}$ (the available periods in the scheduling horizon, one week).
- All lectures have the same duration (say one period)

Output:

An assignment of each lecture L_i to some period in such a way that no student is required to take more than one lecture at a time.

Graph model

Graph $G = (V, E)$:

- V correspond to lectures L_i
- E correspond to conflicts between lectures due to curricula or enrollments

Time slots are colors ➜ Graph-Vertex Coloring problem ➜ NP-complete (exact solvers max 100 vertices)

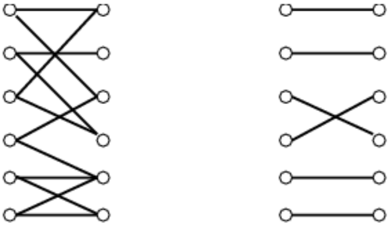
Typical further constraints:

- Preassignments
- Unavailabilities

The overall problem can still be modeled as Graph-Vertex Coloring. How?

A recurrent sub-problem in Timetabling is Matching

Input: A (weighted) bipartite graph $G = (V, E)$ with bipartition $\{A, B\}$.
Task: Find the largest size set of edges $M \in E$ such that each vertex in V is incident to at most one edge of M.



Efficient algorithms for constructing matchings are based on augmenting paths in graphs. An implementation is available in Networkx and:
http://www.cs.sunysb.edu/~algorith/implement/bipm/implement.shtml

Theorem

Theorem [Hall, 1935]: G *contains a matching of* A *if and only if* $|N(U)| \geq |U|$ *for all* $U \subseteq A$.

IP model

Considering indistinguishable rooms:
m_t rooms $\Rightarrow$ maximum number of lectures in time slot t

Variables

$$x_{it} \in \{0, 1\} \qquad i = 1, \ldots, n; \; t = 1, \ldots, p$$

Number of lectures per course

$$\sum_{t=1}^{p} x_{it} = \ell_i \qquad \forall i = 1, \ldots, n$$

Number of lectures per time slot

$$\sum_{i=1}^{n} x_{it} \leq m_t \qquad \forall t = 1, \ldots, p$$

Number of lectures per time slot (students' perspective)

$$\sum_{C_i \in S_j}^{n} x_{it} \leq 1 \qquad \forall j = 1, \dots, n;\; t = 1, \dots, p$$

If some preferences are added:

$$\max \sum_{i=1}^{p} \sum_{i=1}^{n} d_{it} x_{it}$$

Corresponds to a bounded coloring. [de Werra, 1985]

Further complications:

- Teachers that teach more than one course
  (not really a complication: treated similarly to students' enrollment)

- A set of rooms $\mathcal{R} = \{R_1, \ldots, R_n\}$
  with eligibility constraints
  (this can be modeled as Hypergraph Coloring [de Werra, 1985]:
  - introduce an (hyper)edge for events that can be scheduled in the same room
  - the edge cannot have more colors than the rooms available of that type)

Moreover,

- Students' fairness
- Logistic constraints: no two adjacent lectures if at different campus
- Max number of lectures in a single day and changes of campuses.
- Precedence constraints
- Periods of variable length

IP approach

3D IP model including room eligibility [Lach and Lübbecke, 2008]

$R(c) \subseteq \mathcal{R}$: rooms eligible for course c
$G_{conf} = (V_{conf}, E_{conf})$: conflict graph (vertices are pairs (c, t))

$$\min \sum_{ctr} d(c,t) x_{ctr} \qquad \forall c \in \mathcal{C}$$

$$\sum_{\substack{t \in T \\ r \in R(c)}} x_{ctr} = \ell(c) \qquad \forall c \in \mathcal{C}$$

$$\sum_{c \in R^{-1}(r)} x_{ctr} \leq 1 \qquad \forall t \in T, r \in \mathcal{R}$$

$$\sum_{r \in R(c_1)} x_{c_1 t_1 r} + \sum_{r \in R(c_2)} x_{c_2 t_2 r} \leq 1 \qquad \forall ((c_1, t_1)(c_2, t_2)) \in E_{conf}$$

$$x_{ctr} \in \{1, 0\} \qquad \forall (c,t) \in V_{conf}, r \in \mathcal{R}$$

This 3D model is too large in size and computationally hard to solve

2D IP model including room eligibility [Lach and Lübbecke, 2008]

Decomposition of the problem in two stages:

Stage 1 assign courses to timeslots

Stage 2 match courses with rooms within each timeslot
solved by bipartite matching

Model in stage 1

Variables: course c assigned to time slot t

$$x_{ct} \in \{0,1\} \qquad c \in \mathcal{C}, t \in \mathcal{T}$$

Edge constraints
(forbids that c_1 is assigned to t_1 and c_2 to t_2 simultaneously)

$$x_{c_1,t_1} + x_{c_2,t_2} \leq 1 \qquad \forall\, ((c_1,t_1),(c_2,t_2)) \in E_{conf}$$

Hall's constraints
(guarantee that in stage 1 we find only solutions that are feasible for stage 2)
$G_t = (\mathcal{C}_t \cup \mathcal{R}_t, E_t)$ bipartite graph for each t
$G = \cup_t G_t$

$$\sum_{c \in U}^{n} x_{ct} \leq |N(U)| \qquad \forall\, U \subseteq \mathcal{C}, t \in \mathcal{T}$$

If some preferences are added:

$$\max \sum_{i=1}^{p} \sum_{i=1}^{n} d_{it} x_{it}$$

- Hall's constraints are exponentially many
- [Lach and Lübbecke, 2008] study the polytope of the bipartite matching and find strengthening conditions

  (polytope: convex hull of all incidence vectors defining subsets of $\mathcal{C}$ perfectly matched)
- Algorithm for generating all facets is polynomial if the number of defining $\mathcal{C}$-sets is polynomially bounded.
- Could solve the overall problem by branch and cut (separation problem is easy).
  However the number of facet inducing Hall inequalities is in practice rather small hence they can be generated all at once

So far feasibility.

Preferences (soft constraints) may be introduced [Lach and Lübbecke, 2008b]

- Minimum working days
- Curriculum Compactness For every curriculum, the corresponding courses should take place consecutively over a day.
- Room stability
- Student min max load per day
- Travel distance
- Room eligibility
- Double lectures
- Professors' preferences for time slots

Different ways to model them exist.
Often the auxiliary variables have to be introduced

Course/Exam Timetabling

By substituting events with lecture or exam we have the course or exam timetabling, respectively

Differences

Course Timetabling	Exam Timetabling
limited number of time slots	unlimited number of time slots, seek to minimize
conflicts in single slots, seek to compact	conflicts may involve entire days and consecutive days, seek to spread
one single course per room	possibility to set more than one exam in a room with capacity constraints
lectures have fixed duration	exams have different duration