A Column Generation Approach for the post Enrolment Course Timetabling Problem of the ITC

John van den Broek and Cor Hurkens

June 18 2008

Input

- set E of events.
- set T of timeslots (5 days of 9 hours each).
- set R of rooms with $\forall r \in R$:
  - $C_r =$ seating capacity of room r.
  - $F_r =$ set of features satisfied by room r.
- set S of students with $\forall s \in S$:
  - set E_s of events that student s is attending.

Input

$\forall e \in E$:

- F_e = set of features required by event e.
- N_e = number of students attending event e.
- T_e = set of available timeslots for event e.
- R_e = set of allowed rooms for event e.
  - $F_e \subseteq F_r$
  - $N_e \leq C_r$

Precedence requirements:
$\forall e, f \in E : p_{ef} = 1$ if event e has to be scheduled before event f, zero otherwise.

Hard Constraints:

1. No student can attend more than one event at the same time.
2. An event e can only be assigned to a room $r \in R_e$.
3. Only one event is assigned to each room in any timeslot.
4. An event e can only be assigned to a time slot $t \in T_e$.
5. Events have to be scheduled in the prescribed order in the week.

Soft Constraints:

1. Events/students should not be assigned in the last timeslot of a day.

2. Students should not have to attend three or more events in successive timeslots on the same day.

3. Students should not be required to attend only one event a day.

How to compare solutions?

Valid timetable	$\longrightarrow$	no hard constraint violations, unplaced events allowed.
Feasible timetable	$\longrightarrow$	no hard constraint violations and all events in timetable

The quality of solutions is evaluated with two measures:

1. Distance to feasibility (dtf)
2. Total number of violated soft constraints.

Collisions

Two events *collide* if they have:

- a student in common,
- only one possible room that is the same,
- a precedence relation between the two events

c_e = the number of events colliding with event e.

Slot-schedule

A *slot-schedule* k has a timeslot t_k and a set E_k of events.

A slot-schedule is feasible if:
- $\forall e, f \in E_k : e, f$ are not colliding.
- $\forall e \in E_k : t_k \in T_e$.
- $\forall e \in E_k :$ event e is assigned to a room $r \in R_e$.
- At most one event is assigned to each room.

$$a_{ke} := \begin{cases} 1 & \text{if } e \in E_k \\ 0 & \text{otherwise} \end{cases}$$

Room Assignment Generator (RAG)

Input: a set E^p of events and corresponding weights $w_e \in E^p$.

Goal: Determine feasible room assignments with the sum of the weights of the assigned events maximized.

Constraints: Events are assigned to allowed rooms.
Only one event is assigned to each room
No two events are colliding.

Output: a set of feasible room assignments for events in E^p.

Heuristic for RAG

Sort events in:
1. Decreasing order of w_e.
2. Increasing order of c_e.
3. Increasing order of $|R_e|$.

To generate the $p - th$ room assignment:

1. Select event e on position p of the sorted list of events.
2. If $\exists r \in R_e$ that has no event assigned, then assign e to room r and go to 5.
3. Try to find an augmenting path.
4. If augmenting path found, then assign all events to the rooms found in the matching. Otherwise event e can not be assigned.
5. If there are rooms and events left, p:= p+1, go back to 1.

Master Problem

$$\min \sum_{e \in E} y_e + \sum_{e,f \in E | p_{ef}=1} z_{ef}$$

$$y_e + \sum_{k \in K} a_{ke} x_k \geq 1 \qquad \forall e \in E \tag{1}$$

$$\sum_{k \in K | t_k = t} x_k \leq 1 \qquad \forall t \in T \tag{2}$$

$$z_{ef} + \sum_{k \in K} t_k (a_{kf} - a_{ke}) x_k \geq 1 \qquad \forall e, f \in E | p_{ef} = 1 \tag{3}$$

$$x_k \geq 0 \qquad \forall k \in K \tag{4}$$

$$y_e \geq 0 \qquad \forall e \in E \tag{5}$$

$$z_{ef} \geq 0 \qquad \forall e, f \in E | p_{ef} = 1 \tag{6}$$

The Pricing Problem

Weighting factor $w_e, \forall e \in E^p$ is equal to:

$$w_e \;:=\; \begin{cases} (\alpha_e - \gamma_{ef}) & \exists f \in E | p_{ef} = 1 \\ (\alpha_e + \gamma_{fe}) & \exists f \in E | p_{fe} = 1 \\ \alpha_e & \nexists f \in E | p_{ef} = 1 \end{cases}$$

Then the value of the generated column $(= c^k)$is:

$$c^k = \sum_{e \in E^p} w_e y'_e + \beta_t$$

The Column Generation Procedure

1. Initialize period p and the set of columns and set $t = 0$.
2. Solve RMP $\rightarrow$, $\alpha_e, \beta_t, \gamma_{ef}$ (shadowprices) and obj^{rmp}.
   If $obj^{rmp} \leq 0$, then quit.
3. Generate columns for timeslots $t, \ldots, t + p - 1$.
4. Add k if c^k is larger than 0.85 times the average reduced costs over the last 40 added slot-schedules.
5. $t = t + p(mod|T|)$
6. If no new slot-schedules found for a number of periods, then quit.
7. Go to step 2.

Heuristic based on LP-solution

1. Initialize $T_c = T \backslash \{8, 17, 26, 35, 44\}$ and $E_c = E$.
2. Apply column generation procedure $\rightarrow K$.
3. Solve MP as IP, break after five seconds if no optimal solution found.
4. $\forall k \in K, obj_k = \sum_{e \in E} a_{ke} c_e$ - penalties.
5. Fix slot-schedule k' with maximum obj_k.
6. $T_c = T_c \backslash t_{k'}$, $E_c = E_c \backslash E_{k'}$ and delete all columns that are infeasible.
7. If $|T_c| > 0$ and $|E_c| > 0$, then go to step 2.
8. If $|E_c| > 0$, then solve an IP to assign as much as possible of the events in E_c to $t \in \{8, 17, 26, 35, 44\}$.

Computational Results

I	|E|	$\|R\|$	$\|F\|$	$\|S\|$	c.t.(s)	dtf	3 e in row	1 e a day	eod e	s.c.
1	400	10	10	500	316	0	1882	34	508	2424
2	400	10	10	500	324	0	1755	38	529	2322
3	200	20	10	1000	55	0	850	776	0	1626
4	200	20	10	1000	57	0	884	700	0	1584
5	400	20	20	300	209	0	1026	24	213	1263
6	400	20	20	300	218	0	1111	26	232	1369
7	200	20	20	500	29	0	387	321	0	708
8	200	20	20	500	44	0	428	330	0	758
9	400	10	20	500	328	0	1928	33	735	2696
10	400	10	20	500	331	0	1621	38	730	2389
11	200	10	10	1000	43	0	939	713	0	1652
12	200	10	10	1000	64	0	960	552	306	1818
13	400	20	10	300	200	0	1182	31	223	1436
14	400	20	10	300	215	0	1013	13	249	1275
15	200	10	20	500	73	0	497	338	108	943
16	200	10	20	500	40	0	553	340	0	893

Comparison with the best results of the finalists

I	d1	s.c.1	d2	s.c.2	d3	s.c.3	d4	s.c.4	d5	s.c. 5	d us	s.c. us
1	0	61	0	571	0	1482	0	1482	0	1861	0	2424
2	0	547	0	993	0	1635	0	1755	39	2174	0	2322
3	0	382	0	164	0	288	0	850	0	272	0	1626
4	0	529	0	310	0	385	0	884	0	425	0	1584
5	0	5	0	5	0	559	0	1026	0	8	0	1263
6	0	0	0	0	0	851	0	1111	0	28	0	1369
7	0	0	0	6	0	10	0	387	0	13	0	708
8	0	0	0	0	0	0	0	428	0	6	0	758
9	0	0	0	1560	0	1947	0	1928	162	2733	0	2696
10	0	0	0	2163	0	1741	0	1621	161	2697	0	2389
11	0	548	0	178	0	240	0	939	0	263	0	1652
12	0	869	0	146	0	475	0	960	0	804	0	1818
13	0	0	0	0	0	675	0	1182	0	285	0	1436
14	0	0	0	1	0	864	0	1013	0	110	0	1275
15	0	379	0	0	0	0	0	497	0	5	0	943
16	0	191	0	2	0	1	0	553	0	132	0	893

Conclusions

- The heuristic finds a feasible timetable for all instances.

- The number of violated soft constraints is large in comparison with the 5 finalists.

www.cs.qub.ac.uk/itc2007

Wake UP!