Hybrid CP+LNS for the Curriculum-Based Course Timetabling Problem

Tommaso Urli

Dept. of Electrical, Management and Mechanical Engineering
University of Udine, Via Delle Scienze, 206 - 33100 Udine, Italy
tommaso.urli@uniud.it

Abstract. Course Timetabling (CTT) [11] is a popular combinatorial optimization problem, which deals with generating university timetables by scheduling weekly lectures, subject to conflicts and availability constraints, while minimizing costs related to resources and user discomfort. In CB-CTT, students enrol to *curricula* i.e., (possibly overlapping) collections of courses. Lectures of courses pertaining to the same curriculum must therefore be scheduled at different times, so that students can attend all courses. In this paper we propose (i) a Constraint Programming (CP) model for CB-CTT and (ii) a Large Neighborhood Search (LNS) [10] strategy exploiting constraint propagation to solve hard instances of the problem. We then compare our findings against the current best results on the ITC2007 [7] testbed.

1 Introduction

The problem of generating good timetables arises periodically in every university of the world. The problem is inherently complex, involving scheduling weekly lectures while avoiding conflicts i.e., situations in which students would have to attend two lectures at the same time, and possibly generating timetables which cause the least discomfort to both teachers and students. From a theoretical point of view, the underlying graph-coloring problem is NP-complete [6]. For this reason, CB-CTT has been seldom tackled by means of incomplete methods.

Because of its practical relevance, the problem has been widely studied (see [11] for a survey) and many formulations exist, reflecting policies from various universties. Two of them are particularly popular: the Post-Enrolment Course Timetabling (PE-CTT) and the Curriculum-Based Course Timetabling (CB-CTT) [7]. Both formulations have been the subject of the Second International Timetabling Competition (ITC2007) [7], and are currently considered as *standard* formulations. The primary difference between the two formulations is the origin of the matrix of conflicts between lectures. In PE-CTT, students enrol to individual courses, thus two courses are in conflict if they appear together in a student's plan. Conversely, in CB-CTT, students enrol to curricula (presets of courses), so a course is only in conflict with other courses in the same curriculum. In this paper we are focusing on the latter formulation, whose variants have been described thoroughly in [3]. Except for one case, CB-CTT variants only differ on the nature and the relative importance (weight) of soft constraints.

2 Problem description

We now formally define the considered variant of CB-CTT. The definition involves the following entities

- **Days $\times$ Timeslots = Periods.** We are given a set $\mathbf{D}$ of teaching *days*, each one partitioned in a set of *timeslots* $\mathbf{T} \subseteq \mathbb{N}$. Each $p \in \mathbf{P} = \mathbf{D} \times \mathbf{T}$ defines a *period* which is unique within a week.
- **Rooms.** We are allowed to schedule lectures in a set $\mathbf{R}$ of *rooms*, each one with a specific *capacity* $\mathbf{k_r}$ for $r \in \mathbf{R}$. Additionally, a *roomslot* $rs \in \mathbf{RS} = \mathbf{R} \times \mathbf{P}$ represents a room in a specific period.
- **Courses.** A course $c \in \mathbf{C}$ is composed of a set $\mathbf{L_c}$ of *lectures*, that must be scheduled at different times. Each course is taught by a *teacher* $\mathbf{t}_c$ and is attended by a set of *students* $\mathbf{S_c}$. In addition, the lectures of a course should be scattered over a *minimum number of working days* $\mathbf{w_c}$ and must not be scheduled in any period $u \in \mathbf{U_c} \subset \mathbf{P}$ declared as unavailable. The complete set of lecture is $\mathbf{L} = \bigcup \mathbf{L_c}$ for all $c \in \mathbf{C}$.
- **Curricula.** Courses are organized in *curricula* $q \in \mathbf{Q}$ that students can enrol to. Each curriculum has a set of courses $\mathbf{C_q} \subseteq \mathbf{C}$. Lectures pertaining courses in the same curriculum cannot be scheduled together, in order to allow students of each curricula to attend all courses.

A **feasible** solution of the problem is an assignment of a *period* and a *room*, to each lecture, that satisfies the following **hard** constraints

- **Lectures.** All lectures $\mathbf{L}$ must be scheduled.
- **Room occupancy.** Two lectures $l_1, l_2 \in \mathbf{L}, l_1 \neq l_2$ cannot take place in the same roomslot, no matter what course or curriculum they pertain to.
- **Conflicts.** Lectures in the same course or in conflicting courses i.e., same teacher or same curriculum, may not be scheduled at the same time.
- **Availabilities.** A course may not be taught in any of its unavailable periods.

Feasible solutions can be ranked on the basis of their violations of the following **soft constraints**

- **Room capacity.** Every lecture $l \in \mathbf{L_c}$ for $c \in \mathbf{C}$ should be scheduled in a room $r \in \mathbf{R}$ so that $\mathbf{k_r} \leq |\mathbf{S_c}|$, which can accomodate all of its students. If this is not the case, $|\mathbf{S_c}| - \mathbf{k_r}$ *room capacity violations* are considered.
- **Room stability.** Every lecture $l \in \mathbf{L_c}$ for $c \in \mathbf{C}$ should be given in the same room. Every additional room used for a lecture in course c generates a *room stability violation.*
- **Minimum working days.** Lectures $\mathbf{L_c}$ of a course $c \in \mathbf{C}$ should be scattered over a minimum number of working days $\mathbf{w_c}$. Each time a course is scheduled in $d < \mathbf{w_c}$ days, we count $\mathbf{w_c} - d$ *working days violations.*
- **Isolated lectures.** When possible, lectures of the same curriculum should be adjacent within a day. Each time a lecture is not preceded or followed by lectures of courses in the same curriculum counts as a *isolated lectures violation.*

Thus, in addition of being feasible, a solution s should have the minimal linear combination of soft violations (see weights in Table 1)

$$cost(s) = rc(s) \cdot w_{rc} + rs(s) \cdot w_{rs} + mwd(s) \cdot w_{mwd} + il(s) \cdot w_{il} \quad (1)$$

Violation	**Weight**	**Symbol**
Room capacity	1	rc
Room stability	1	rs
Minimum working days	5	mwd
Isolated lectures	2	il

Table 1. Weights of the various types of violations [3]

3 Related work

We discuss previous work involving both CP models for other CTT variants, and meta-heuristics for CB-CTT. To the best of our knowledge, our approach is the first to address CB-CTT from a CP perspective.

Constraint programming. Cambazard *et al.* [4] describe a hybrid approach for PE-CTT, that shares many similarities with our technique. Among other things, the authors propose a CP model, coupled with a LNS search strategy, to tackle complex instances of the problem. Some interesting insight is given on the approach. First, the authors stress the importance of releasing the right variables during the LNS step. Second, they propose a Simulated Annealing (SA) acceptance criterion to escape local minima. Third, they handle feasibility and optimization in different search phases. Cipriano *et al.* [5] describe a framework for the integration of a CP solver with LNS and solve a simple CTT variant where conflicts are solely determined by the availability of teachers.

Meta-heuristics. Many meta-heuristic methods have been proposed to deal with the CB-CTT problem. Müller [9] describes a multi-phase local-search algorithm combining a constructive forward search to obtain a feasible solution with successive local-search steps based on Hill-Climbing (HC), Great Deluge (GD) and SA. Bellio *et al.* [2] propose a hybrid local-search algorithm which alternates SA with dynamic Tabu Search (TS) with shifting penalties. The work is supported by an extensive statistical analysis on the effect of parameters. Lü and Hao [8] propose a three-phase hybrid algorithm which improves a greedy feasible solution through alternate intensification (TS) and perturbation. The trade-off between intensification and diversification is controlled by two parameters that are adjusted based on the past performance. Abdullah *et al.* [1] describe a hybrid meta-heuristic based on GD and an electromagnetic-like mechanism (EM) that performed very well on various tracks of the ITC2007 competition.

4 Model and search strategy

Similarly to [4] and [5], we model CB-CTT in CP, and explore it employing a LNS [10] search strategy. For the purpose of reproducibility, both the model and of the LNS engine are available at `http://bitbucket.org/tunnuz/cpctt`.

4.1 Model

Variables. We represent a solution as a set of $roomslot_l$ variables, that represent both the room and the period a lecture $l \in \mathbf{L}$ is scheduled in. The variable domains are initialized as $dom(roomslot_l) = \{1 \ldots |\mathbf{R}| \cdot |\mathbf{P}|\}$. Additionally, some redundant variables (namely day_l, $period_l$, $timeslot_l$, $room_l$ with the obvious channellings) are used as modeling sugar.

Hard constraints. Since we are using exactly $|\mathbf{L}|$ decision variables, expressing the **Lectures** and **Room occupancy** constraints is trivial

$$\texttt{alldifferent}(roomslot)$$

To model the **Conflicts** constraint, we must take into account pairs of conflicting courses and constrain their lectures to be scheduled at different periods

$$\texttt{alldifferent}(\{roomslot_l \mid l \in \mathbf{L_{c_1}} \cup \mathbf{L_{c_2}}\})$$

$\forall\, c_1, c_2 \in \mathbf{C}, \texttt{conflicting}\,(c_1, c_2)$ where `conflicting` checks whether two courses belong to the same curriculum or have the same teacher (note that this holds when checking a course against itself). Finally, the **Availabilities** constraint can be modeled by imposing that

$$period_l \notin \mathbf{U_c}, \quad \forall\, c \in \mathbf{C}, l \in \mathbf{L_c}$$

Note that, for our purpose, some of the hard constraints are available both as hard and soft constraints. In particular, the **Lectures** and **Conflicts** constraint can be transformed into soft constraints by imposing `nvalues` and `count` constraints between the *roomslot* and *period* variables and two auxiliary variables, and then embedding the auxiliary variables in the cost function.

Soft constraints. For each of the soft constraints, we defined an auxiliary variable to accumulate the violations of the constraints. For some of them, such as **Room stability**, this involves counting, for each $c \in \mathbf{C}$ how many different values (`nvalues`) were taken by $\{roomslot_l \mid l \in \mathbf{L_c}\}$, and then subtracting it from $|L_c|$ to calculate how many extra rooms were used by the course. A similar approach was taken to compute the violations for **Minimum working days**, while set variables and `cardinality` constraints were used for **Isolated lectures**.

4.2 Search strategy

While some instances can be solved by CP alone, this is not the case for the harder ones. In particular, the essential **Lectures** and **Conflicts** hard constraints are very difficult to tackle with classic *branch & bound.*

Our approach treats them as soft constraints, and solves the lexicographic minimization problem where first all hard violations are eliminated, and then the objective function (Equation 1) is minimized. In order to get an advantage on instances where CP alone is able to solve the hard constraints, we start each resolution with a 10-seconds *branch & bound* attempt to solve the complete model.

Our LNS search strategy revolves around the following ideas

- **SA-like cost bounding.** In order to escape from local minima, we allow the cost of a neighboring solution to increase. However, to avoid disrupting the positive effect that cost bounding has on propagation, we reverse the standard SA probability of acceptance, and we compute the allowed cost increase as $\Delta = -(t \ln p)$, where t is the typical temperature parameter and $p \sim \mathcal{U}(0, 1)$ is a random number in $[0, 1]$. Temperature t is updated as $t = t \cdot \lambda$, with $0 < \lambda < 1$ after ρ solutions have been accepted at the current level of t.
- **Biased relaxation.** At each LNS step we select a number $d \in \{d_{min}, d_{max}\}$ of variables to release, and re-assign them through *branch & bound*. A fraction of the variables to release is chosen heuristically based on to the constraints being violated by the solution, the rest is chosen randomly. Branch & bound is given a timeout of $t_{var} \cdot d$ milliseconds where t_{var} is a constant.
- **Growing neighborhoods and stagnation.** After $iter_{max} \cdot d$ iterations have been spent at a certain d without any improvement, d is increased in a VNS fashion. When $d = d_{max}$ the search restarts with a perturbed solution.
- **Adaptive d_{min}.** At each restart, the new d_{min} is set to the d that yielded the highest number of (temporary) best solutions in the past iterations.

In general, our biased relaxation mechanism works as follows: whenever a lecture l causes some violations (either hard or soft), the variable $roomslot_l$ and the variables related to the other lectures involved in the violation, are released. As for the parameters, we currently use $d_{min} = 2$, $d_{max} = 5\%$ of the number of lectures, $t_{var} = 10$, $t_{init} = 35$, $\rho = 5$, $iter_{max} = 250$ and $\lambda = 0.97$.

5 Results

Table 2 shows our preliminary results (grey) against the current best ones in literature, on the ITC2007 testbed. Our approach is still outperformed by the well-established algorithms on many instances and further investigation is needed to better deal with local minima in order to improve our solutions.

6 Conclusions and future work

Coupling a LNS strategy with a CP model seems a promising way to get the best of the two worlds. On the one hand, CP allows to model problems in a high-level language and comes with powerful propagation techniques. On the other hand, LNS allows to explore the search space while exploiting the neighborhood reduction provided by propagation. We expect the importance of neighborhood reduction to become more and more relevant when dealing with problems involving large neighborhoods or combination of them.

One of the most promising directions for the future of this approach is the use of multiple channeled CP models to provide even better propagation and make the LNS step faster. Another interesting area for development is the acceptance criterion which, in our opinion, is fundamental to deal with local minima.

Inst.	**Müller [7] ∪ [9]**		**Lü & Hao [8]**		**Abdullah *et al.* [1]**		**Bellio *et al.* [2]**		**CP+LNS**		**Best**
	avg	best	avg	best	avg	best	avg	best	median	best	
1	**5.0**	**5**	**5.0**	**5**	**5.0**	**5**	**5.00**	**5**	6.0	5	5
2	61.3	43	60.6	**34**	53.90	39	**53.0**	40	219.5	158	24
3	94.8	72	86.6	**70**	84.20	76	**79.0**	**70**	226.0	158	66
4	42.8	**35**	47.9	38	51.90	**35**	**38.3**	**35**	92.0	62	35
5	343.5	**298**	**328.5**	**298**	339.5	315	365.20	326	931.5	637	290
6	56.8	**41**	69.9	47	64.40	50	**50.4**	**41**	174.0	130	27
7	33.9	14	28.2	19	**20.20**	**12**	23.8	17	156.5	97	6
8	46.5	39	51.4	43	47.90	**37**	**43.6**	40	162.5	70	37
9	113.1	103	113.2	99	113.90	104	**105.0**	**98**	216.0	173	96
10	21.3	**9**	38.0	16	24.10	10	**20.5**	11	137.5	91	4
11	**0.0**	**0**	**0.0**	**0**	**0.0**	**0**	**0.00**	**0**	0.0	0	0
12	351.6	331	365.0	**320**	355.90	337	**340.5**	325	716.0	616	300
13	73.9	66	76.2	65	72.40	**61**	**71.3**	64	152.0	120	59
14	61.8	**53**	62.9	52	63.30	**53**	**57.9**	54	131.0	103	51
15	94.8	–	87.8	**69**	88.00	73	**78.8**	70	226.5	150	66
16	41.2	–	53.7	38	51.70	32	**34.8**	**27**	124.5	93	18
17	86.6	–	100.5	80	86.20	72	**75.7**	**67**	198.5	152	56
18	91.7	–	82.6	**67**	85.80	77	**80.8**	69	144.5	116	62
19	68.8	–	75.0	**59**	78.10	60	**67.0**	61	199.0	141	57
20	**34.3**	–	58.2	35	42.90	**22**	38.8	33	185.0	137	4
21	108.0	–	125.3	105	121.50	95	**100.1**	**89**	257.5	209	75

Table 2. Comparison with the best approaches in literature on ITC2007 instances. Timeout (5 minutes) has been calculated using the competition benchmarking tool.

References

1. S. Abdullah, H. Turabieh, B. McCollum, and P. McMullan. A hybrid metaheuristic approach to the university course timetabling problem. *Journal of Heuristics*, 18(1):1–23, 2012.
2. R. Bellio, L. Di Gaspero, and A. Schaerf. Design and statistical analysis of a hybrid local search algorithm for course timetabling. 15(1):49–61, 2012.
3. A. Bonutti, F. De Cesco, L. Di Gaspero, and A. Schaerf. Benchmarking curriculum-based course timetabling: formulations, data formats, instances, validation, visualization, and results. 194(1):59–70, 2012.
4. H. Cambazard, E. Hebrard, B. O'Sullivan, and A. Papadopoulos. Local search and constraint programming for the post-enrolment-based course timetabling problem. 2008.
5. R. Cipriano, L. Di Gaspero, and A. Dovier. A multi-paradigm tool for large neighborhood search. In *Hybrid Metaheuristics*, volume 434 of *Studies in Computational Intelligence*, pages 389–414. Springer Verlag, 2012.
6. D. de Werra. An introduction to timetabling. 19:151–162, 1985.
7. L. Di Gaspero, B. McCollum, and A. Schaerf. The second international timetabling competition (ITC-2007): Curriculum-based course timetabling (track 3). Technical report, Queen's University, Belfast (UK), August 2007.
8. Z. Lü and J.-K. Hao. Adaptive tabu search for course timetabling. 200(1):235 – 244, 2009.
9. T. Müller. ITC2007 solver description: a hybrid approach. 172(1):429–446, 2009.
10. D. Pisinger and S. Ropke. Large neighborhood search. In *Handbook of metaheuristics*, pages 399–419. Springer, 2010.
11. A. Schaerf. A survey of automated timetabling. *Artificial Intelligence Review*, 13(2):87–127, 1999.