CrossMark

Comments on: An overview of curriculum-based course timetabling

Marco E. Lübbecke

A search on google scholar for *university timetabling* reveals not only a wealth of material. There are also a number of publications which refer to *survey* or *overview* in their title, including the recent ones by Kristiansen and Stidsen (2013) and Babaeia et al. (2014).

One could ask whether it is necessary to add another one to the stock, and I say yes, this overview was much needed. As timetabling is so hard for exact approaches to cope with, the literature mainly dealt with heuristic, in particular meta-heuristic solution algorithms. Only recently, integer programming based methods entered the scene, and actually quite successfully so. Also, the formulations based on satisfiability problems are very interesting. It is in particular because of the presentation of these (in principle) exact approaches why I like this overview.

It is an *overview* indeed and this is what I would like to mainly address in this comment: It would be an even more valuable document if the authors added a little more advise for those who would like to actually *do* timetabling.

1 Theoretical vs. practical timetabling

The competition induced by the ITC challenges certainly stimulated a broad and deep development in timetabling. Yet, the theoretical problem definition, expressed in Sect. 2, may be quite far from timetabling reality. Even those constraints ensuring that every lecture (a) must be scheduled, (b) in a conflict-free manner (both considered as irrevocably hard) may be impossible to keep in the age of complex interdependencies

---

M. E. Lübbecke (✉)
Operations Research, RWTH Aachen University, Kackertstr. 7, 52072 Aachen, Germany
e-mail: marco.luebbecke@rwth-aachen.de

of many interdisciplinary curricula at large universities. If I could make a wish, it would be the next overview written on the subject to treat the practical, may be even the *dirty* side of university timetabling.

2 Multiple objectives and robustness

The authors briefly touch upon multiple, often conflicting objectives in university timetabling. Working with *soft constraints* is the community's classical reaction to this, penalizing the violation of constraints by appropriately chosen weights in the (single) objective function. It happens easily that ten or more of such penalty terms become relevant in practice.

From my own experience with practical optimization problems, and university timetabling in particular, these many "apples and oranges" in the objective function somewhat obscure the evaluation of a solution, making different timetables hard to compare. Worse, the penalties are often of different orders of magnitude, solutions are computed only up to a certain quality (not to optimality), and as a result, different runs with (slightly) different data yield significant structural differences. This adds to hard comparability in practice; moreover extreme changes to timetables are hard to communicate in certain environments.

The computation of all Pareto-optimal solutions, and more importantly: their presentation to the user is practically hardly feasible and offers only little amends to the above situation.

This said, from a theoretical and practical point of view, *robustness* and *stability* of timetables are an important and timely topic. With their great experience in robust optimization, the authors are encouraged to provide valuable input to the community also in this respect in the future.

3 Computational results

Fair computational comparisons of different approaches are hard to obtain, as the authors state themselves. The ITC challenges provided a reasonable environment to come very close to fairness, but this does not necessarily apply to the publications reporting about the various approaches. The settings, machines, and allowed computation times vary significantly between the papers discussed in the overview, and I was wondering how the comparison would be if all approaches were run under identical conditions. This part of the overview needs to be taken with a grain of salt.

4 Perspectives and challenges

When we presented our *two-stage approach* (Lach and Lübbecke 2008, 2012) (it actually is a Benders' decomposition) and went from a *three-indexed* integer programming formulation to a *two-indexed* one, we were intrigued by the idea of formulating a *one-indexed* model. The formulation by Cacchiani et al. (2013) is such a model.

I believe that column generation and branch-and-price is still worthwhile further and deeper investigation, and this overview article supports this, but this perception may be personally biased.

I would thus appreciate if the authors could give their account on what approaches they find most promising for achieving which goals in timetabling.

Going a little further, the overview wants to "open new perspectives of further developments". It would be nice to see these perspectives, in particular the scientific ones, more clearly. The extensions presented in Sect. 5, and others mentioned in the literature or only encountered in practice yet, are important but can be (and are) modeled directly, e.g., in integer programs. Does this imply that these extensions are already there? In this context, it is a strength of the overview, and of the timetabling community at large, that the availability of test instances and standardized problem formulations ensures better comparability between approaches. On the other hand, timetabling in practice is very diverse, and advise about which approaches work best (computationally) in which situations would be a useful addition for practitioners. I understand that such advise can hardly be given without experimenting with all the models and approaches oneself. Yet, besides an overview of existing material a (personal) account of the suitability for which purposes would be appreciated.

Let me challenge the authors to providing more *vision* in their rejoinder where they see the field in several years.

5 Yet more to come!

There was a third international timetabling competition, ITC-2011, which dealt with highschool timetabling (Post et al. 2013), so the community on university course timetabling, or CB-CTT, did not compete for a while. I would like to particularly see how integer programming approaches fare in comparison to the meta-heuristic state-of-the-art.

This overview certainly inspired me to look into our old code, install a fresh CPLEX and see what is possible in 2015. While solvers dramatically improved, the challenges still lie on the modeling and algorithmic side. How can we exploit special problem structures? Looking very much forward to seeing new ideas.

It appears to be about time for a new ITC challenge in university course timetabling!

References

Babaeia H, Karimpourb J, Hadidic A (2014) A survey of approaches for university course timetabling problem. Comput Ind Eng. doi:10.1016/j.cie.2014.11.010

Cacchiani V, Caprara A, Roberti R, Toth P (2013) A new lower bound for curriculum-based course timetabling. Comput Oper Res 40(10):2466–2477. doi:10.1016/j.cor.2013.02.010

Kristiansen S, Stidsen T (2013) A comprehensive study of educational timetabling, a survey. In: Report 8.2013, DTU Management Engineering, Denmark's Technical University (DTU)

Lach G, Lübbecke M (2008) Optimal university course timetables and the partial transversal polytope. In: McGeoch C (ed) Proceedings of the 7th Workshop on Experimental Algorithms (WEA), vol 5038, pp 235–248. Springer, Berlin. doi:10.1007/978-3-540-68552-4_18

Lach G, Lübbecke M (2012) Curriculum based course timetabling: new solutions to Udine benchmark instances. Ann Oper Res 194(1):255–272. doi:10.1007/s10479-010-0700-7

Post G, Di Gaspero L, Kingston JH, McCollum B, Schaerf A (2013) The third international timetabling competition. Ann Oper Res 1–7. doi:10.1007/s10479-013-1340-5