Instructor Scheduling in UniTime

Tomáš Müller

Purdue University, West Lafayette, Indiana, USA
muller@unitime.org

1 Introduction

Instructor scheduling is the problem of assigning instructors to classes subject to various constraints, such as instructor availability, qualifications, maximal load, teaching preferences, etc. While this problem can be included in the course timetabling component (instructors are assigned together with class times and rooms), instructors are usually assigned before or after the course timetabling [10]. Typically, professors or lecturers are already known when the course timetable is being built, and avoiding instructor clashes is a constraint commonly present in course timetabling problems [9]. On the other hand, teaching assistants and additional instructors are assigned after the class schedule has been published.

UniTime[1] is an open-source tool for university course [9] and examination [4] time-tabling, student [7] and instructor scheduling, and event management. While we have published several papers on various components over the years, the problem of assigning instructors has been neglected. It should also be noted that real-world timetabling data collected using UniTime has been used for the ITC 2019 competition [8], and a version of the UniTime solver has been used by the author in the ITC 2007 competition [3], where it was among the finalists in all three tracks and won two of them. While the same solver is used for instructor scheduling as for the other components, the problem model differs, with some interesting and challenging constraints.

The work on this module has been motivated by the teaching assistant assignments at Purdue University. Purdue University is a large public university with over 40 thousand students at its main campus in West Lafayette, Indiana, USA. Many departments, such as Mathematics, Chemistry, or Biology, offer large service courses attended by hundreds of students each semester. These courses usually consist of a small number of large lectures taught by faculty members and dozens of small lab sections, each manned by a teaching assistant. At Purdue, the course timetable is usually set about six months in advance, but teaching assistants are only assigned a few weeks before classes start, ensuring they can attend the classes they need for their studies. Additional personnel may need to be hired to cover the teaching load.

Time-wise, instructor scheduling occurs after course timetabling is complete, but it may overlap with student scheduling. This means that any instructor assignments are also considered during student scheduling, preventing a student

[1] https://www.unitime.org

who is also a teaching assistant from making a registration change that would conflict with their teaching schedule. However, in theory, the individual modules can be used in any order, as instructor assignments are also checked for conflicts during course timetabling. This provides a lot of flexibility, especially when dealing with disruptions that occur later in the scheduling process.

In this abstract, we describe the university instructor scheduling problem (UISP) as modeled in UniTime and briefly explain the algorithm used to solve it. For more details about the module, please refer to the UniTime documentation[2].

2 Instructor Scheduling Problem

The UISP consists of teaching requests and instructors. While these are mostly teaching assistants at Purdue, other teaching staff, such as professors and lecturers, can be included in the problem if desired. Each **instructor** has

- a **maximal teaching load**, typically expressed in the number of hours
- **teaching preference**, providing an optional penalization of an instructor if they are assigned a teaching request; this is used, e.g., to penalize outside personnel that would need to be hired, and would represent an additional cost for the department
- a **list of attributes** that can present various qualifications, skills, certifications, or other department-defined criteria that the instructor meets
- **time preferences** that allow an instructor to prohibit certain times and/or prefer/discourage some other times; time preferences are provided for individual intervals during the week, e.g., preferred Mondays 8 AM - noon, prohibited Fridays
- **course preferences** that allow an instructor to require, prefer, discourage, or prohibit certain courses
- **distribution preferences** that allow an instructor to provide preferences or requirements about the distribution of their class assignments, such as preferring back-to-back classes or requiring the same room for all their teaching assignments

As with other modules, UniTime uses a seven-level scale for preferences and requirements: **prohibited** and **required** preference levels are treated as hard constraints and must always be satisfied by the solver. All other preference levels have associated penalties: **strongly discouraged** (+4), **discouraged** (+1), **neutral** (0), **preferred** (-1), and **strongly preferred** (-4).

A **teaching request** represents a class that needs an instructor assigned to it. A single teaching request may contain multiple classes, as a course may have lab-recitation pairs that require the same instructor, or the instructor may need to be present at the corresponding lecture as well, which is common to multiple teaching requests. A teaching request has

- a **teaching load** expressing how many hours the instructor will need

[2] https://help.unitime.org/manuals/instructor-scheduling

- **one or more classes** of a course that the instructor would teach; along with their scheduled times and rooms, if already timetabled
- optional **common class or classes** of the course during which the instructor also needs to be available; typically, a part of the course common to multiple teaching requests, such as a lecture
- **attribute preferences** that include requirements and preferences for various attributes, such as qualifications, skills, certifications, etc.
- **instructor preferences** that allow for individual instructors to be preferred, discouraged, prohibited, or required
- **same-course** and **same-common** preference

Instructor attributes and teaching request attribute preferences allow modeling a fairly complex relationship between classes that need an instructor assigned (teaching requests) and instructors available to teach those classes or courses. For example, certain qualifications may be required, while other skills or certifications are preferred or discouraged. The design allows attributes to be grouped by type, and each type may specify whether its attributes are conjunctive or disjunctive. If multiple attributes of the same type are required on a teaching request, an instructor may need to have all the required attributes (conjunctive) or just one of them (disjunctive). This allows complete customization of attribute types and individual attributes.

The same-course and the same-common preferences arise from using the UISP primarily to assign teaching assistants. Typically, a teaching assistant has two, three, or four teaching assignments that must be from the same course and are preferred to share a common part (e.g., are associated with the same lecture).

A solution to a UISP is a set of teaching assignments, where each teaching request is assigned to one instructor. While partial solutions are permitted (some teaching requests may be left unassigned), all hard constraints must always be satisfied. Moreover, the aim is to minimize the weighted sum of the various optimization criteria (soft constraints/penalizations).

The problem has the following hard constraints that must always be satisfied

- The **teaching load** of an instructor cannot exceed the instructor's maximal load.
- The instructor **must be available** during their teaching assignment; i.e., there is no overlap with a prohibited time or any class they attend.
- An instructor's **teaching assignments cannot overlap** in time or share the same class, except for the common class.
- Instructors with a **prohibited teaching preference** cannot be used.
- Required and prohibited **attributes**, **instructor**, **course** and **distribution preferences** must be obeyed.
- Required and prohibited **same-course**, **same-common** must be obeyed. E.g., when an instructor has one teaching request that requires the same-course, all their assignments must be for the same course; a prohibited same-common would mean that no other assignment of the instructor shares any of the common classes of the request.

The optimization criteria (soft constraints) include the following criteria. Each represents a sum of the penalizations from negative (-4 for strongly preferred and -1 for preferred) to positive (+1 for discouraged and +4 for strongly discouraged) applied based on whether the preferred/discouraged time, course, instructor, etc., is used:

- **teaching preference**: applies to each teaching assignment of the instructor
- **time preference**: penalization of the worst preference overlapping with the teaching assignment
- **course preference**: penalization of a course preference set on an instructor
- **instructor preference**: penalization of an instructor preference set on a teaching request
- **attribute preference**: sum of penalizations of all attribute preferences that match the instructor attributes
- **distribution preference**: penalization for each violation of a distribution preference
- **unused instructor load** for instructors with one or more assignments; that is, the difference between the maximal load of the instructors and the sum of the loads of the assigned teaching assignments

Each of the above criteria has a weight in the solver configuration. The aim is to find a complete solution that minimizes the objective function, defined as a weighted sum of the above criteria.

3 Algorithm

The presented UISP is modeled as a Constraint Satisfaction Problem, where teaching requests are modeled as variables to be assigned. The domain of a teaching request is a list of instructors who can teach the request, meaning they satisfy the required/prohibited time, course, attribute, and instructor preferences and do not overlap with any class the instructor is attending as a student. This reduces the hard constraints that need to be checked during the search to the maximum load, teaching assignment overlaps, required/prohibited distributions, and required/prohibited same-course and same-common settings among the teaching assignments of the same instructor.

As with the other UniTime modules, the solver is implemented using the CPSolver library [2], sharing the common primitives such as variables, values, constraints, optimization criteria, search algorithms, and general heuristics with the other UniTime problems, and implementing only the problem-specific components. The CPSolver source code is available at GitHub[3], including the instructor scheduling solver.

The search algorithm starts with the iterative forward search (IFS) [6], in which, at each iteration, one unassigned variable is selected and assigned. This can lead to the unassignment of conflicting assignments. The conflict-based

[3] https://github.com/UniTime/cpsolver

statistics (CBS) [5] are used to prevent the algorithm from getting stuck in a local optimum and to provide the user with information when a complete solution cannot be found. Once a complete solution is obtained, the great deluge (GD) [1] technique is used to optimize it. The bound is dynamically set based on the best-known solution and reset when it falls below a threshold. During this phase, only instructor swaps and reassignments that do not violate any hard constraints can be selected, forcing the solver to move from one complete solution to another. Besides simple swaps and reassignments, the search employs a limited-depth branch-and-bound.

4 Conclusion

Among all the UniTime modules, the instructor scheduling module usually receives the least attention. This is because the individual problems are typically very small, involving only teaching assistants and/or other instructors within a single department.

It is possible that instructor scheduling will be integrated with the course timetabling module in the future, perhaps in a similar way to how student sectioning is combined with class assignments to time and space [9, 8].

References

1. Dueck, G.: New optimization heuristics: The great deluge algorithm and the record-to record travel. Journal of Computational Physics **104**, 86–92 (1993)
2. Müller, T.: University course timetabling: Solver evolution. In: Practice and Theory of Automated Timetabling 2016 Proceedings. pp. 263—-282 (2016)
3. Müller, T.: Itc2007 solver description: a hybrid approach. Annals of Operations Research **127**, 429–446 (2009). https://doi.org/10.1007/s10479-009-0644-y
4. Müller, T.: Real-life examination timetabling. Journal of Scheduling **19**, 257–270 (2016). https://doi.org/10.1007/s10479-014-1643-1
5. Müller, T., Barták, R., Rudová, H.: Conflict-based statistics. In: Gottlieb, J., Silva, D.L., Musliu, N., Soubeiga, E. (eds.) EU/ME Workshop on Design and Evaluation of Advanced Hybrid Meta-Heuristics. University of Nottingham (2004)
6. Muller, T., Barták, R., Rudová, H.: Iterative forward search: Combining local search with maintaining arc consistency and a conflict-based statistics. In: LSCS'04 — 1st International Workshop on Local Search Techniques in Constraint Satisfaction (2004)
7. Müller, T., Murray, K.: Comprehensive approach to student sectioning. Annals of Operations Research **181**, 249–269 (2010)
8. Müller, T., Rudová, H., Müllerová, Z.: Real-world university course timetabling at the International Timetabling Competition 2019. Journal of Scheduling **28**, 247–267 (2025). https://doi.org/10.1007/s10951-023-00801-w
9. Rudová, H., Müller, T., Murray, K.: Complex university course timetabling. Journal of Scheduling **14**(2), 187–207 (2011)
10. Xue, G., Offodile, O.F., Razavi, R., Kwak, D.H., Benitez, J.: Addressing staffing challenges through improved planning: Demand-driven course schedule planning and instructor assignment in higher education. Decision Support Systems **187** (2024). https://doi.org/10.1016/j.dss.2024.114345