Metaheuristic optimization of Danish High-School Timetables

Victor Alexander Brandsen[1] [0009-0008-4823-1496] and Thomas Stidsen[2] [0000-0001-6905-5454]

[1] Technical University of Denmark, Kgs. Lyngby 2800, Denmark `s184014@student.dtu.dk`
[2] Technical University of Denmark, Kgs. Lyngby 2800, Denmark `thst@dtu.dk`

Abstract. Creating schedules for high schools are a massive undertaking, with schools having hundreds of students, teachers, rooms, and courses, all with different requirements. Here we briefly explain a methodology to create schedules of 10, 20, and 40 weeks for Danish high schools, including event chains.

Keywords: Scheduling, Metaheuristics, Mathematical Programming.

1 The Danish High School Timetabling Problem

The High School Timetabling Problem (HSTP) is a well-known problem, and even though it is very dependent on the country of the high schools in question, general versions have been formulated [1]. A common part of HSTP is however to have a cyclic schedule, where a fixed timetable is used for the entire school year. Here we will briefly describe the Danish High School Timetabling Problem (DHSTP). The model described is now used by (practically) all Danish high schools and has some unique features. The current model of DHSTP is used for cyclic repetition of the timetable, but here we experiment with using metaheuristics for solving the DHSTP for not one week, but up to 40 weeks.

The DHSTP model utilize both hard and soft constraints, which are shown below in table 1. Metaheuristics and Mixed Integer Programming (MIP) have previously been applied to the DHSTP ([2,3,4]), showing the viability of both methods in scheduling over 1 and 2 weeks. Real-life instances are often too complex for scheduling 10 or more weeks, whereby planning for two weeks and repeating the schedule over the longer horizon is a viable solution. This however means that single instances of resources not being available during the longer horizon cannot be considered, potentially leading to sub-optimal solutions. If, however, creating schedules of 10 weeks or more were viable in practice, more information about the school year can be taken into account since the optimization processes then can re-assign schedule elements as needed.

1.1 Event chains

An important difference between the HSTP and DHSTP is the addition of event chains. Event chains describe a requirement for a series of events to be placed at certain timeslot offsets relative to a shared origin, e.g.:

Table 1: Hard constraints in the DHSTP

	Hard constraints
Placement	Events requires a timeslot
Event locks	Events requiring a specific timeslot/room
Resource conflict	Limits on usage of resources in a timeslot
Availability	Events or resources only being available in certain timeslots
Event chains	Events required to take place in relation to other events
Work limit	Teachers not having more than a set amount of events per day
Days off	Teachers having at least a set amount of days off
Days off stability	Days off each week for a teacher not differing by more than 1
Day conflict	Events in the same course not being assigned to the same day

- Events in the chain have to be assigned the same timeslot.
- Events in the chain must be assigned contiguous timeslots, in order.
- One event in the chain must occur exactly 2 timeslots after another event.

Moving where this shared origin lies within the schedule therefore changes the correct position of the event chain. These are of great practical use, e.g. when planning educational days with a specific theme, but large chains can cause problems for schedulers due to the significant space required in the schedule to place these correctly.

1.2 Objective Function

A MIP-model for the DHSTP has previously been presented [2,4], including the hard constraints described in Table 1. This model is largely applied here, with one significant change: Event chains are changed from a hard constraint to multiple heavily penalized soft constraints.

To do this, the following goals for placing event chains have been identified:

1. Events in event chains are assigned correct timeslots relative to each other
2. Events in event chains are assigned a timeslot

Regarding goal 1: A definition is required for when an event in an event chain is assigned correctly. As seen in Fig. 1, there can be multiple interpretations of the same schedule, potentially leading to different amounts of correctly placed events. This can be alleviated by deciding on a specific event in the event chain (the "root-event" of the chain), which, if placed, the other events must be placed relative to. In Fig. 1, Case 1 corresponds to having the event in timeslot 5 as the root-event, and Case 2 corresponds to having the event in timeslot 1 as the root event. Since the only way to assess if events are correctly placed is if the root-event is placed, a penalty is added on events in event chains without the root-event assigned. Additionally, a penalty is added to incorrectly placed events in the chain when the root-event is placed. These penalties in combination achieve goal 1 and partly goal 2.

Regarding goal 2: This is achieved by penalizing unassigned events in event chains, only when the root-event of the chain is placed, since it is deemed undesirable to have the event chains only partly scheduled.

The objective of the model is therefore to minimize a weighted sum of:

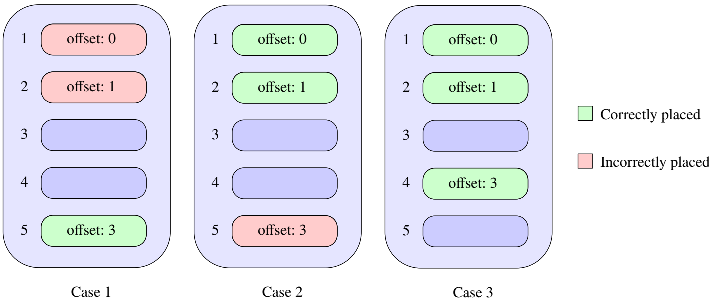

Fig. 1: Two cases scheduling three events in an event chain, on a single day with five timeslots. The required relative offsets for the three events is noted.
Case 1: Event with offset 3 is considered as being correctly placed.
Case 2: Event with offset 0 is considered as being correctly placed.
Case 3: All events are correctly placed.

1. The number of events without a timeslot assigned,
2. The number of events without a room assigned,
3. The index of the timeslots assigned to each event,
4. The number of teachers scheduled at timeslots they wish not to be scheduled on,
5. A desirability-score of the rooms assigned to events,
6. The number of events on neighboring days for courses,
7. The number of idle timeslots for teachers and students,
8. The number of rooms used by each course in excess of one,
9. The number of work days for teachers,
10. The number of days where students have zero events scheduled,
11. The number of days where teachers have exactly one event scheduled,
12. The maximum difference in the number of scheduled events in a week for courses,
13. The number of events in event chains without the root-event placed,
14. The number of incorrectly placed events in event chains with the root-event placed, and
15. The number of unassigned events in event chains with the root-event placed.

2 Solution Methods

A dataset of 20 instances of schedules for 1 and 2 weeks from [2], with between 300 and 2500 events and between 140 and 660 classes, are used. As mentioned, it is of interest to create schedules of 10 or more weeks, but due to the results from the MIP-approach in [2] on 1 and 2 weeks, it is deemed infeasible to find exact solutions to the DHSTP for longer planning horizons, and a metaheuristics-approach is therefore used. Adaptive Large Neighborhood Search (ALNS) and Tabu Search (TS) algorithms are proposed for generating solutions for schedules of 1, 2, 10, 20, and 40 weeks. For schedules of 10 or more weeks, three approaches are furthermore explored:

- A cold-start approach, where a greedy heuristic is applied to empty schedules of 10, 20, and 40 weeks, which are then used as starting points for the metaheuristics.
- A warm-start approach, where schedules for 1 or 2 weeks are created with the cold-start approach, which then are duplicated to create initial schedules of 10, 20, and 40 weeks. Illegally placed events are removed and metaheuristics are again applied.
- A two-stage approach, where timeslots are assigned first and rooms assigned afterward, as outlined in [2]. This approach can be combined with the cold-start or warm-start approaches.

3 A brief conclusion

At the presentation, a summary of the model used and conclusions drawn from the results from using the above methods will be presented.

References

1. Kristiansen, S., Sørensen, M., Stidsen, T.R.: Integer programming for the generalized high school timetabling problem. Journal of Scheduling **18**, 377–392 (2015)
2. Schjørring, N.D.: Mathematical Optimisation of High-School Timetabling. Master's thesis, Technical University of Denmark (2023)
3. Sørensen, M., Stidsen, T.R.: High School Timetabling: Modeling and solving a large number of cases in Denmark. Proceedings of the Ninth International Conference on the Practice and Theory of Automated Timetabling (PATAT) pp. 359–364 (2012)
4. Sørensen, M., Stidsen, T.R.: Integer programming and adaptive large neighborhood search for real-world instances of high school timetabling. Annals of Operations Research, PATAT (2012)