Adaptive Large Neighborhood Search for Student Sectioning at Danish high schools

Simon Kristiansen · Thomas R. Stidsen

Keywords Student Sectioning · Elective Course Planning · Adaptive Large Neighborhood Search · Educational Timetabling · High School Planning · Metaheuristics

Mathematics Subject Classification (2000) 90-08 · 90B35 · 90C10 · 90C59

1 Introduction

Student Sectioning is one of the less studied subjects within Educational Timetabling (Pillay (2010)). Student Sectioning is normally used at universities (e.g. Müller and Murray (2010) and Cheng et al (2003)), whereas this paper is concerned high schools in Denmark. I.e. a more generalized model is needed since this problem should be applicable for approximately 200 different high schools. The concerned problem is also known as Elective Course Planning Problem (ECPP), described in Kristiansen et al (2011b). However in this paper we will also try to pack the students more convenient in the classes. I.e. minimize the number of common classes in each class and to have a more even distribution between classes of same course. The ECPP is a preceding planning problem of the actual High School Timetabling in Denmark. The Internation Timetabling Competition 2011 was devoted to High School Timetabling (see e.g. Post et al (2012) and Sørensen et al (2012)).

The students request some elective courses, and the problem is then to assign course classes to time slots and then assign students to the classes given their requests. We want to maximize the number of fulfilled requests and to minimize the number of classes. The problem is to both please the students and to insure good economic. It cost approximately €27.000 p.a. to create a class, and as the high schools are self-governing, ECPP is a crucial problem. If the requests are not granted, the students might change school and as the high school are payed upon the number of students, they will lose revenue.

	Monday	Tuesday	Wednesday	Thursday	Friday
8:15 9:45	Block1			Block3	
10:00 11:30					
	Lunch break				
12:00 13:30					
13:45 15:15		Block2		Block4	Block5

Fig. 1: *An example of a weekly schedule with four modules each day and a total of five time slots for elective courses. The white time slots are used for mandatory courses.*

Simon Kristiansen [sikr@dtu.dk] · Thomas R. Stidsen
Operation Research, DTU Management, Technical University of Denmark
Produktionstorvet, 426 B, DK-2800, Kgs Lyngby, Denmark

As mentioned it is also desired to have an even distribution on the number of students in classes of same course. E.g. 40 students requests the same given course, and with 28 as upper bound on the class size, we aim at having a distribution of 20-20.

2 Integer programming model

The ECPP is formulated as an IP model. A high school has a set of students $s \in \mathcal{S}$, a set of offered courses $e \in \mathcal{E}$, a set of classes $c \in \mathcal{C}$ and a set of time slots $b \in \mathcal{B}$. Each students is assigned to a common class, $q \in \mathcal{Q}$, this is denoted by $I_{q,s} \in \{0,1\}$. Each course belongs to one of the course subjects given by the set $f \in \mathcal{F}$. The maximum number of classes of each subject in a time slot is given by $M_f \in \mathbb{R}^+$. $K_{c,f} \in \{0,1\}$ denotes whether course c is teaching subject f or not. For each course class there exist an upper bound on the class sizes, $U_c \in \mathbb{R}^+$. $A_{c,s} \in \mathbb{R}^+$ indicates whether student s is locked to course class c. I.e. the student must be assigned to the given course class. $R_{e,s} \in \{0,1\}$ indicates whether student s has requested course e or not. $D_{c,e} \in \{0,1\}$ denotes whether class c is teaching course e, or not. The total maximum number of classes which can be created is given by $P \in \mathbb{R}^+$. The parameters $J_{c,c'} \in \{0,1\}$ and $H_{c,c'} \in \{0,1\}$ indicates whether two classes cannot be placed in the same time slot or should be placed in the same time slot, respectively. The decision whether student s is assigned to class c in block b is defined by $x_{c,b,s} \in \{0,1\}$, while the decision whether course class c is assigned to time slot b is given by $y_{c,b} \in \{0,1\}$. The binary variable $z_{c,q} \in \{0,1\}$ takes value 1 if common class q is present in class c. The variables $w^+_{c,c'} \in \mathbb{N}_0$ and $w^-_{c,c'} \in \mathbb{N}_0$ counts the difference of the number of students between classes of same course. The objectives are weighted in respect to each other, given by $\alpha_{c,s}$, β_c, γ and δ

$$
\begin{aligned}
\max \quad & \sum_{c,b,s} \alpha_{c,s} \cdot x_{c,b,s} - \sum_{c,b} \beta_c \cdot y_{c,b} - \gamma \cdot \sum_{c,q} z_{c,q} - \delta \cdot \sum_{c,c'} (w^+_{c,c'} + w^-_{c,c'})/2 && && (1)\\
\text{s.t.} \quad & \sum_{c} x_{c,b,s} && \leq 1 && \forall\, b, s && (2)\\
& \sum_{b} y_{c,b} && \leq 1 && \forall\, c && (3)\\
& \sum_{c}\sum_{b} x_{c,b,s} \cdot D_{c,e} && \leq R_{e,s} && \forall\, e, s && (4)\\
& \sum_{s} x_{c,b,s} && \leq U_c && \forall\, c, b && (5)\\
& \sum_{b}\sum_{s} I_{q,s} \cdot x_{c,b,s} && \leq z_{c,q} && \forall\, c, q, \sum_{s} A_{c,s} = 0 && (6)\\
& \sum_{s} x_{c,b,s} - \sum_{s} x_{c',b,s} && = w^+_{c,c'} - w^-_{c,c'} && \forall\, c, c', b, e, D_{c,e} = D_{c',e} = 1, \sum_{s} A_{c,s} = \sum_{s} A_{c',s} = 0 && (7)\\
& x_{c,b,s} && \leq y_{c,b} && \forall\, c, b, s, A_{c,s} = 0 && (8)\\
& x_{c,b,s} && = y_{c,b} && \forall\, c, b, s, A_{c,s} = 1 && (9)\\
& \sum_{c,b} y_{c,b} && \leq P && && (10)\\
& y_{c,b} + y_{c',b} && \leq 1 && \forall\, c, c', b, s, J_{c,c'} = 1 && (11)\\
& y_{c,b} && = y_{c',b} && \forall\, c, c', b, s, H_{c,c'} = 1 && (12)\\
& \sum_{c} K_{c,f} \cdot y_{c,b} && \leq M_f && \forall\, b, f, M_f > 0 && (13)\\
& x_{c,b,s} \in \{0,1\} && && && (14)\\
& y_{c,b} \in \{0,1\} && && && (15)\\
& z_{c,q} \in \{0,1\} && && && (16)\\
& w^+_{c,c'} \in \mathbb{N}_0 && && && (17)\\
& w^-_{c,c'} \in \mathbb{N}_0 && && &&
\end{aligned}
$$

The constraints (2) ensure that no students is assigned more than one course class in each time slot, while constraints (3) ensure that a class cannot be assigned to more than one time slot. Constraints (4) ensure that students can only be assigned course classes which they requested, and that each requests is only granted once. Constraints (5) sets the upper bound on the number of students in a course class. Constraints (6) are counting the number of common classes used while constraints (7) are used for equal distribution of the students in classes of same course. Constraints (8) is the connection between the two variables $x_{c,b,s}$ and $y_{c,b}$ and make sure that a student cannot be assigned a class which is not yet assigned to a time slot. Constraints (6) and (7) are soft constraints. Constraints (9) shall ensure that if students are locked to a course class and the class is assigned a timeslot, the students should be assigned to the class. Constraints (10) ensures that the total number of created classes does not exceed maximum. Constraints (11) ensure that classes which cannot be placed in same time slot are not done so and constraints (12) ensure that classes which should be placed in same time slot are satisfied. Finally, constraints (13) make sure that the resource limit on subjects f are respected.

3 Solution method

It has been chosen to attempt Adaptive Large Neighborhood Search (ALNS) to establish solutions to the ECPP. ALNS was first applied and is still mainly used on Vehicle Routing Problems (Azi et al (2010); Laporte et al (2010); Salazar-Aguilar et al (2011); Ribeiro and Laporte (2012)). It has however also been applied on a few other problems such as Lot-sizing (Muller and Spoorendonk (2011)) and Scheduling problems (Muller (2010); Kristiansen et al (2011a); Sørensen and Stidsen (2012)). Pisinger and Ropke (2010) is recommended for additional reading on ALNS. The pseudo code is given in Algorithm 1.

Algorithm 1: Adaptive Large Neighborhood Search

Input: a feasible solution $x^s_{g,b}$, $q \in \mathbb{N}$
solution $x_{best} = x$; $\pi = (1, \ldots, 1)$
repeat
    $x' = x$
    select destroy and repair methods $d \in \Omega^-$ and $r \in \Omega^+$ using π
    remove q requests from x' using d
    reinsert removed requests into x' using r
    if $c(x') > c(x_{best})$ **then**
        $x_{best} = x'$
    if *accept*(x', x) **then**
        $x = x'$
    update π
until *stop-criterion met*
return x_{best}

The neighborhoods are implicitly defined by several destroy and repair methods. In each iteration, a destroy and a repair method is chosen upon some performance indicators which is updated after each iteration.

For this implementation of ALNS two different types of moves are used: Assign/unassign a class with students to/from a time slot and assign/unassign a student to/from an assigned class. Based on these moves a total of 8 different destroy methods and 4 repair methods are implemented. The destroy methods are simple random removal heuristics and Shaw heuristics (Shaw (1997)), where less or more related classes are removed from the solution. The repair methods are basic greedy algorithms and regret heuristics (Potvin and Rousseau (1993)), which aims at inserting the request which we will regret most if not inserted immediately.

4 Results

The algorithm is implemented in Lectio [1], and is hence available for use for approximately 200 different high schools in Denmark. This gives the possibilities for a huge amount of data for further testing and research. Table 1 shows the size and the computational results from 7 different datasets

	No.of student	No.of requests	No.of courses	No.of blocks	Assigned classes	Assigned requests
Vejen	382	586	29	3	36	586
Silkeborg	927	1789	65	5	77	1786
Falkoner	421	1080	49	4	66	1080
Vordingborg	415	1462	61	5	68	1462
Alssund	385	650	31	5	34	645
Holstebro	345	567	18	5	29	567
Frederikssund	159	273	18	4	18	273

Table 1: *Results for a given set of real-life problems at Danish high schools.*

References

Azi N, Gendreau M, Potvin JY (2010) An Adaptive Large Neighborhood Search for a Vehicle Routing Problem with Multiple Trips. CIRRELT

Cheng E, Kruk S, Lipman M (2003) Flow formulations for the student scheduling problem. In: Burke E, De Causmaecker P (eds) Practice and Theory of Automated Timetabling IV, Lecture Notes in Computer Science, vol 2740, Springer Berlin / Heidelberg, pp 299–309

Kristiansen S, Sørensen M, Stidsen T, Herold M (2011a) Adaptive large neighborhood search for the consultation timetabling problem, to appear

Kristiansen S, Sørensen M, Stidsen TR (2011b) Elective course planning. European Journal of Operational Research 215(3):713 – 720, DOI 10.1016/j.ejor.2011.06.039

Laporte G, Musmanno R, Vocaturo F (2010) An adaptive large neighbourhood search heuristic for the capacitated arc-routing problem with stochastic demands. Transportation Science 44(1):125–135

Müller T, Murray K (2010) Comprehensive approach to student sectioning. Annals of Operations Research 181:249–269

Muller L (2010) An adaptive large neighborhood search algorithm for the multi-mode resource-constrained project scheduling problem. Tech. rep., Department of Management Engineering, Technical University of Denmark Produktionstorvet, Building 426, DK-2800 Kgs. Lyngby, Denmark

Muller L, Spoorendonk S (2011) A hybrid adaptive large neighborhood search algorithm applied to a lot-sizing problem. European Journal of Operational Research Volume 218(Issue 3):614–623

Pillay N (2010) An overview of school timetabling research. In: Proceedings of the International Conference on the Theory and Practice of Automated Timetabling, Belfast, United Kingdom, pp 321–335

Pisinger D, Ropke S (2010) Large neighborhood search. In: Gendreau M, Potvin JY (eds) Handbook of Metaheuristics, International Series in Operations Research & Management Science, vol 146, Springer US, pp 399–419

Post G, Gaspero LD, Kingston JH, McCollum B, Schaerf A (2012) The third international timetabling competition. In: Proceedings of the Ninth International Conference on the Practice and Theory of Automated Timetabling (PATAT 2012), Son, Norway

Potvin JY, Rousseau JM (1993) A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. European Journal of Operational Research 66(3):331 – 340

Ribeiro GM, Laporte G (2012) An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem. Computers & Operations Research 39(3):728 – 735, DOI 10.1016/j.cor.2011.05.005

Salazar-Aguilar M, Langevin A, Laporte G (2011) An adaptive large neighborhood search heuristic for a snow plowing problem with synchronized routes. In: Pahl J, Reiners T, Voss S (eds) Network Optimization, Lecture Notes in Computer Science, vol 6701, Springer Berlin / Heidelberg, pp 406–411

Shaw P (1997) A new local search algorithm providing high quality solutions to vehicle routing problems

Sørensen M, Stidsen TR (2012) High school timetabling: Modeling and solving a large number of cases in denmark. In: Proceedings of the Ninth International Conference on the Practice and Theory of Automated Timetabling (PATAT 2012)

Sørensen M, Kristiansen S, Stidsen TR (2012) International timetabling competition 2011: An adaptive large neighborhood search algorithm. In: Proceedings of the Ninth International Conference on the Practice and Theory of Automated Timetabling (PATAT 2012)

[1] Lectio is developed by MaCom A/S and is a cloud-based high school administration, which handles all sorts of administrative tasks for the high schools, including a GUI and a heuristic-based solver for the ECPP.