J Heuristics (2013) 19:465–495
DOI 10.1007/s10732-013-9219-9

The consultation timetabling problem at Danish high schools

Simon Kristiansen · Matias Sørensen · Michael B. Herold · Thomas R. Stidsen

Received: 23 June 2011 / Revised: 13 September 2012 / Accepted: 31 March 2013 /
Published online: 24 April 2013

Abstract In the different stages of the educational system, the demand for efficient planning is increasing. This article treats the $\mathcal{NP}$-hard *Consultation Timetabling Problem*, a recurrent planning problem for the high schools in Denmark, which has not been described in the literature before. Two versions of the problem are considered, the *Parental Consultation Timetabling Problem* (PCTP) and the *Supervisor Consultation Timetabling Problem* (SCTP). It is shown that both problems can be modeled using the same Integer Programming model. Solutions are found using the state-of-the-art MIP solver Gurobi and *Adaptive Large Neighborhood Search* (ALNS), and computational results are established using 300 real-life datasets. These tests show that the developed ALNS algorithm is significantly outperforming both Gurobi and a currently applied heuristic for the PCTP. For both the PCTP and the SCTP, it is shown that the ALNS algorithm in average provides results within 5 % of optimum. The developed algorithm has been implemented in the commercial product Lectio, and is therefore available for approximately 95 % of the Danish high schools.

Keywords High school · Timetabling · Metaheuristics · Integer programming · Adaptive Large Neighborhood Search · F-Race tuning

Mathematics Subject Classification (2000) 90-08 · 90C10 · 90C30 · 90B35 · 90C59

S. Kristiansen (✉) · M. Sørensen · T. R. Stidsen
Section of Operations Research, Department of Management Engineering, Technical University of Denmark, Building 426, Produktionstorvet, 2800 Kgs. Lyngby, Denmark
e-mail: sikr@dtu.dk

S. Kristiansen · M. Sørensen · M. B. Herold
MaCom A/S, Vesterbrogade 48, 1, 1620 Copenhagen V, Denmark

1 Introduction

The *Consultation Timetabling Problem* (CTP) is a recurrent planning problem for the high schools in Denmark, which concerns the creation of a schedule for student-teacher meetings, given the students requests of teachers, subject to various soft constraints and resource constraints. The problem has not been described in the literature before, but it shares some properties with other problems within the educational sector, see Sect. 2.3. There exists several variants of the problem. In this paper we consider the two most important versions for the Danish high schools, namely the *Parental Consultation Timetabling Problem* (PCTP) and the *Supervisor Consultation Timetabling Problem* (SCTP).

This paper is written in cooperation with the Danish company MaCom A/S. MaCom A/S is the developer of the cloud-based high school administration system Lectio, which handles all sorts of administrative tasks for the high schools, including a GUI and a heuristic-based solver for the PCTP. Through this cooperation we have access to real-life data for approximately 95 % of all Danish high schools, which constitutes thousands of datasets. We will provide computational results for 300 of these datasets, which is a very big amount of real-life data compared to the majority of scheduling literature.

Our task of this paper is to give a detailed description of the CTP, and model it using Integer Programming. This model should support both the PCTP and the SCTP. To find solutions, both a state-of-the-art MIP solver and an *Adaptive Large Neighborhood Search* (ALNS) heuristic is attempted. These solution approaches are compared to the existing heuristic used in Lectio, and the best approach is made available for all users of Lectio.

2 Consultation timetabling problem

In the following we describe the CTP in details, starting with specifications of the two versions of the problem.

2.1 Parental consultation timetabling problem

Once or twice a year the high schools invite the students and their parents to participate in meetings with the teachers of the student. The goal of these meetings is to allow the teachers to inform of the educational progress of the student, and possibly address relevant problems. Parental consultations usually take place in the evening of a normal work day, and each meeting generally has a duration between 5 and 15 min. The order of events for parental consultations is the following: The high school administration selects days where the meetings should take place, and for each day a feasible time-interval is chosen. Each student (usually in collaboration with his parents) makes prioritized requests of groups of teachers he would like to meet. Few of these teacher groups contain more than a single teacher, because the student is taught by only one teacher in most classes. Usually the high school also allows the students to request specific time intervals, within the overall time interval on each day, where the student will be available for meeting teachers. Given the student's choice of teachers and time intervals, it is then up to the high school administration to decide which teachers a student should meet, and when the meetings should take place.

2.2 Supervisor consultation timetabling problem

In the last year of a high school education, the students are required to make a large study project (Danish: *Studieretnings Projekt*). Each student selects two subjects/courses as combined subject for his project, e.g. English and History. Each student is then assigned two teachers whom will be his supervisors for the project. The objective of the SCTP is to plan meetings between the students and their respective supervisors. The goal of these meetings is for the supervisors to provide the student with some useful hints for problem definition, literature research, etc. Typically supervisor consultations take place in the daytime, where both the student and the corresponding teachers are located at the high school.

From a timetabling point of view, these two types of consultations are almost identical. For both types, as many as possible of the meeting requests should be fulfilled, and both essentially contain the same constraints. Therefore we will in this paper model both types of consultations using the same Integer Programming model. In the remainder of this paper we refrain, as much as possible, from distinguishing between the two variants of the problem, and will by CTP denote the problem which is the superset of the PCTP and the SCTP.

In the following further details of the soft constraints of the CTP is given. These soft constraints define various scheduling preferences for the students and teachers.

A contiguous streak of meetings for a teacher or student are from now on denoted a *sequence*. A time slot is *void* for a teacher or student if the time slot is empty and no meetings are scheduled in either earlier time slots or in later time slots. A *break* for a student or teacher is defined as a time slot which is not void, and which has no meetings assigned. Void time slots must be distinguished from breaks because they do not effect the density of a schedule. This is due to the fact that students and teachers are not obligated to stay at the school throughout the entire duration of the consultation. Given these definitions, we formulate the following soft constraints:

- It is attempted to achieve a solution where the positions of the granted meetings for a given individual are placed in a sequence. I.e. for both students and teachers the number of breaks should be minimized. This is to achieve a schedule with as little waiting time as possible. However, for the students it is possible for the high school administration to declare whether they need a break after each meeting. This is usually selected if there exists ”traveling” time between the meeting rooms where the teachers are located. The CTP only takes consultation meetings into consideration when determining a sequence, and not other activities.
- When assigning a meeting to a time slot, the availability of the student and teacher must be taken into consideration. The high school administration decides whether this constraint should be defined as a hard- or a soft-constraint. It is common that in case of SCTP, this is defined as a soft-constraint as it is feasible for the students to leave classes to have meetings with their supervisors. In case of the PCTP,

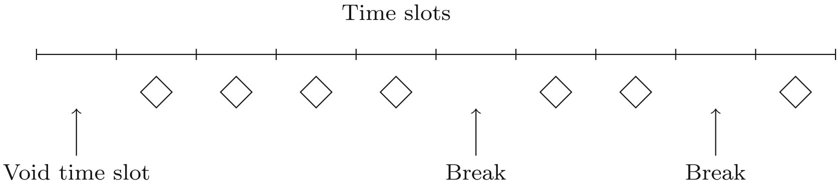

Fig. 1 Example of a feasible consultation schedule

this constraint is usually treated as a hard-constraint, as a solution should respect activities such as meetings, study-trips, etc.

- It is undesirable for teachers to have too long sequences of meetings. Therefore a maximum on the length of sequences for teachers is given (treated as a soft-constraint). This is not required for the students, since they typically have a low number of requests.
- For a consultation which spans over several days it is desired that a student or a teacher only have meetings in one of the days, such that they are not obligated to attend both. This is especially critical for students, as they have a low number of requests.
- The high school administration prefers if the meetings are placed as close as possible to a specific time slot on each day. This is usually selected as the first timeslot on each day.

Figure 1 shows an example of a consultation schedule on one day. The schedule contains one void time slot, two breaks, and seven consultations.

2.3 Literature review

The CTP has, to the best of our knowledge, not been described in the literature before. However there exist many types of related timetabling problems within the educational system, see Schaerf (1999), Burke and Petrovic (2002), McCollum (2006), and Pillay (2010) for overviews of this field. Problems such as *Course Timetabling* and *Student Sectioning* have been looked into, e.g. Tripathy (1984), Erben and Keppler (1996), and Carter and Laporte (1998), Müller and Murray (2010). Related problems for Danish high schools include Kristiansen et al. (2011) and Kristiansen and Stidsen (2012) regarding the *Elective Course Planning Problem*, and Sørensen and Stidsen (2012) regarding the timetabling problem. For all these problems it applies that they attempt to assign requests to time slots in a given schedule.

The requirement for compact schedules is well known in educational timetabling. In Santos et al. (2010) the Class-Teacher Timetabling Problem with Compactness Constraints is described. The compactness is defined in terms of teacher "holes", which is equivalent to our definition of breaks. The teacher holes are modeled with a linear IP, which entails the need for two auxiliary variables, and three additional constraints. This approach can be directly applied to the CTP. de Haan et al. (2007) specifies that the timetabling problem at Dutch high schools requires compact schedules for both classes and teachers. This is addressed using heuristic methods. For Greek high schools the situation is similar, see e.g. Birbas et al. (2009). Here the problem is handled using a MIP solver for a complicated IP.

3 Integer programming model

The following IP model for the CTP aims at maximizing the number of granted meeting requests and minimizing the violating of soft constraints, while respecting the hard constraints.

A consultation problem at a high school contains set of students S, a set of teachers T, a set of teacher groups G, an ordered set of time slots B and a set of days D. $V_{b,d} \in \{0, 1\}$ takes value 1 if time slot b is part of day d, and zero otherwise.

The decision whether a student s is given a meeting with teacher group g in time slot b is defined by the binary variable $x_{s,g,b} \in \{0, 1\}$. The profit of meeting (s, g) in timeslot b is given by $\alpha_{s,g,b} \in \mathbb{R}^+$. The basic objective function is therefore

$$\max \sum_{s,g,b} \alpha_{s,g,b} x_{s,g,b} \tag{1}$$

3.1 Unavailabilities

In some situations it can be allowed to interrupt other activities at the high schools to satisfy a meeting request. Let $D_{t,b} \in \{0, 1\}$ take value 1 if teacher t is not available (i.e. occupied by other activities) in time slot b, and zero otherwise. Let $E_{s,b} \in \{0, 1\}$ be the completely analogous parameter for the students. If a consultation meeting is placed in a time slot where either a student or a teacher has some other activities, it is penalized by the following.

$$-\sum_{s,g,b} \left(\sum_{t} \delta_t \cdot D_{t,b} \cdot P_{g,t} + \delta_s \cdot E_{s,b} \right) x_{s,g,b} \tag{2}$$

If it is not allowed to interrupt activities this term is not added to the objective, and these constraints take the form of hard-constraints by forbidding meetings of teacher t in time slot b if $D_{t,b} = 1$, and likewise for student s in time slot b if $E_{s,b} = 1$.

It is of course not allowed to assign a student to a consultation meeting with a teacher group, if the student has not requested this teacher group. And is it not allowed for the student or teacher to have more than one meeting in each time slot. This imposed the following constraints. The parameter $P_{t,g} \in \{0, 1\}$ takes value 1 if teacher t is in teacher group g, and zero otherwise. $R_{s,g} \in \{0, 1\}$ takes value 1 if student s has requested teacher group g, and zero otherwise. $C_{s,b} \in \{0, 1\}$ takes value 1 if student s has requested time slot b.

$$\sum_{b} x_{s,g,b} \leq R_{s,g} \qquad \forall s, g \tag{3}$$

$$\sum_{g} x_{s,g,b} \leq C_{s,b} \qquad \forall s, b \tag{4}$$

$$\sum_{s,g} P_{g,t} \cdot x_{s,g,b} \leq 1 \qquad \forall t, b \tag{5}$$

3.2 Undesirable breaks

One of the undesirable properties of the CTP is the breaks for both students and teachers. Let the variables $z_{s,d} \in \mathbb{N}$ and $w_{t,d} \in \mathbb{N}$ be the number of breaks in day d for a student and for a teacher, respectively. As we do not penalize void time slots as shown in Fig. 1, we need to know when a individual have his first and last meeting. Let variables $f^{\text{first}}_{s,d}$ and $f^{\text{last}}_{s,d}$ denote the timeslot of the first and last meeting for student s, respectively. Let $h^{\text{first}}_{t,d}$ and $h^{\text{last}}_{t,d}$ be the analogous variables for teacher t. Let variable $n_{s,b,d} \in \{0, 1\}$ take value 1 if student s is placed in time slot b on day d. The idle time slots for a student is then given by the following constraints

$$\sum_{g} V_{b,d} \cdot x_{s,g,b} = n_{s,b,d} \qquad \forall s, b, d, V_{b,d} = 1 \tag{6}$$

$$f^{\text{last}}_{s,d} - f^{\text{first}}_{s,d} + 1 - \sum_{b} n_{s,b,d} \cdot (1 + HS) + HS = z_{s,d} \qquad \forall s, d \tag{7}$$

$$|B|_d - (|B|_d - \text{ord}(b)) \cdot n_{s,b,d} \geq f^{\text{first}}_{s,d} \qquad \forall s, b, d, V_{b,d} = 1 \tag{8}$$

$$\text{ord}(b) \cdot n_{s,b,d} \leq f^{\text{last}}_{s,d} \qquad \forall s, b, d, V_{b,d} = 1 \tag{9}$$

The parameter $HS \in \{0, 1\}$ indicates whether a student is required a break after each meeting, zero otherwise. This parameter is selectable for the user of Lectio. We want to penalize the cost such that it increases exponential on the number of breaks. The cost function is modeled as a piece-wise linear function by introducing a new variable $vs_{s,d,j} \in \{0, 1\}$, where $j \in 1, \ldots, m$, which takes value 1 if student s has j breaks in day d. This imposes the following constraints.

$$\sum_{j} vs_{s,d,j} \cdot \text{ord}(j) = z_{s,d} \qquad \forall s, d \tag{10}$$

$$\sum_{j} vs_{s,d,j} = 1 \qquad \forall s, d \tag{11}$$

As for the teacher let variable $p_{t,b,d} \in \{0, 1\}$ take value 1 if teacher t has a meeting in time slot b on day d. The cost for teacher breaks is also made as a piece-wise linear function, using the new variable $vt_{t,d,j} \in \{0, 1\}$. The following constraints are imposed to denote the number of undesirable breaks for a given teacher,

$$\sum_{g,s} V_{b,d} \cdot P_{g,t} \cdot x_{s,g,b} = p_{t,b,d} \qquad \forall t, b, d, V_{b,d} = 1 \tag{12}$$

$$|B|_d - (|B|_d - \text{ord}(b)) \cdot p_{t,b,d} \geq h^{\text{first}}_{t,d} \qquad \forall t, b, d, V_{b,d} = 1 \tag{13}$$

$$\text{ord}(b) \cdot p_{t,b,d} \leq h^{\text{last}}_{t,d} \qquad \forall\, t, b, d, V_{b,d} = 1 \tag{14}$$

$$h^{\text{last}}_{t,d} - h^{\text{first}}_{t,d} + 1 - \sum_{b} p_{t,b,d} = w_{t,d} \qquad \forall\, t, d \tag{15}$$

$$\sum_{j} vt_{t,d,j} \cdot \text{ord}(j) = w_{t,d} \qquad \forall\, t, d \tag{16}$$

$$\sum_{j} vt_{t,d,j} = 1 \qquad \forall\, t, d \tag{17}$$

The contribution to the objective is as follows

$$-\sum_{s,d,j} \gamma_{s,j} \cdot vs_{s,d,j} - \sum_{t,d,j} \beta_{t,j} \cdot vt_{t,d,j} \tag{18}$$

3.3 Sequences

In connection with the undesirable breaks, the CTP also contains some needed breaks. For the students it is often necessary to give them a break between each consultation meeting due to traveling time between meeting rooms. This impose the following constraints.

$$\sum_{g} \left(x_{s,g,b} + x_{g,b_{+1},s} \right) \leq 1 \;\forall\, s, d, b \in B \backslash \{b_{|B_J|}\}, HS = 1, V_{b,d} = V_{b_{+1},d} = 1 \tag{19}$$

The teachers seldom change location during the consultation period, so travel time is not needed. However, as mentioned it is undesirable for the teachers to have a long sequence of meetings, as they need a break now and then. The maximum size of a sequence for a teacher is denoted $Q \in \mathbb{N}$. Let the variable $y_{t,b,d} \in \{0, 1\}$ take value 1 if time slot b is the start of a sequence of length greater than Q on day d for teacher t. The following equation constraints this variable,

$$\sum_{s} \sum_{\substack{b'=b \\ V_{b',d}=1}}^{b+Q} p_{t,b',d} - y_{t,b,d} \leq Q \qquad \forall\, t, d, b \in B \backslash \{b_j | j > |B| - Q\}, V_{b,d} = 1 \tag{20}$$

The contribution to the objective is given by

$$-\sum_{t,b,d} \omega \cdot y_{t,b,d} \tag{21}$$

3.4 Day distribution

In case the consultation has more than one day it is preferred that each student and each teacher only has meetings on a single day. $u_{t,d} \in \{0, 1\}$ and $u_{s,d} \in \{0, 1\}$ denotes if teacher t or student s has a meeting on day d, respectively. $v_t \in \mathbb{N}$ and $v_s \in \mathbb{N}$ denotes the number of days where teacher t and student s have meetings, respectively. The number of days with meetings is punished in the objective by

$$-\sum_s \zeta_s \cdot v_s - \sum_t \zeta_t \cdot v_t \tag{22}$$

and is constrained by the following

$$\sum_g V_{b,d} \cdot x_{s,g,b} \leq u_{s,d} \qquad \forall s, b, d \tag{23}$$

$$\sum_{s,g} V_{b,d} \cdot P_{t,g} \cdot x_{s,g,b} \leq u_{t,d} \qquad \forall t, b, d \tag{24}$$

$$\sum_d u_{s,d} - 1 \leq v_s \qquad \forall s \tag{25}$$

$$\sum_d u_{t,d} - 1 \leq v_t \qquad \forall t \tag{26}$$

The entire model for CTP is given in (27).

3.5 IP model for CTP

$$\textit{Consultation Timetabling Problem IP} \tag{27}$$

$$\max \sum_{s,g,b} \left(\alpha_{s,g,b} - \sum_t \delta_t \cdot D_{t,b} \cdot P_{g,t} + \delta_s \cdot E_{s,b} \right) \cdot x_{s,g,b} - \sum_{s,d,j} \gamma_{s,j} \cdot vs_{s,d,j} - \sum_{t,d,j} \beta_{t,j} \cdot vt_{t,d,j}$$

$$- \sum_{t,b,d} \omega \cdot y_{t,b,d} - \sum_s \zeta_s \cdot v_s - \sum_t \zeta_t \cdot v_t \tag{27a}$$

$$\text{s.t.} \sum_b x_{s,g,b} \leq R_{s,g} \qquad \forall s, g \tag{27b}$$

$$\sum_g x_{s,g,b} \leq C_{s,b} \qquad \forall s, b \tag{27c}$$

$$\sum_{s,g} P_{g,t} \cdot x_{s,g,b} \leq 1 \qquad \forall t, b \tag{27d}$$

$$\sum_g \left(x_{s,g,b} + x_{g,b+1,s} \right) \leq 1 \qquad \forall s, d, b \in B \backslash \{b_{|B_J|}\}, HS = 1, V_{b,d} = V_{b+1,d} = 1 \tag{27e}$$

$$\sum_s \sum_{\substack{b'=b \\ V_{b',d}=1}}^{b+Q} p_{t,b',d} - y_{t,b,d} \leq Q \qquad \forall t, d, b \in B \backslash \{b_j | j > |B| - Q\}, V_{b,d} = 1 \tag{27f}$$

$$\sum_{g} V_{b,d} \cdot x_{s,g,b} = n_{s,b,d} \qquad \forall s, b, d, V_{b,d} = 1 \tag{27g}$$

$$|B|_d - (|B|_d - \mathrm{ord}(b)) \cdot n_{s,b,d} \geq f_{s,d}^{\mathrm{first}} \qquad \forall s, b, d, V_{b,d} = 1 \tag{27h}$$

$$\mathrm{ord}(b) \cdot n_{s,b,d} \leq f_{s,d}^{\mathrm{last}} \qquad \forall s, b, d, V_{b,d} = 1 \tag{27i}$$

$$f_{s,d}^{\mathrm{last}} - f_{s,d}^{\mathrm{first}} + 1 - \sum_{b} n_{s,b,d} \cdot (1 + HS) + HS = z_{s,d} \qquad \forall s, d \tag{27j}$$

$$\sum_{j} vs_{s,d,j} \cdot \mathrm{ord}(j) = z_{s,d} \qquad \forall s, d \tag{27k}$$

$$\sum_{j} vs_{s,d,j} = 1 \qquad \forall s, d \tag{27l}$$

$$\sum_{g,s} V_{b,d} \cdot P_{g,t} \cdot x_{s,g,b} = p_{t,b,d} \qquad \forall t, b, d, V_{b,d} = 1 \tag{27m}$$

$$|B|_d - (|B|_d - \mathrm{ord}(b)) \cdot p_{t,b,d} \geq h_{t,d}^{\mathrm{first}} \qquad \forall t, b, d, V_{b,d} = 1 \tag{27n}$$

$$\mathrm{ord}(b) \cdot p_{t,b,d} \leq h_{t,d}^{\mathrm{last}} \qquad \forall t, b, d, V_{b,d} = 1 \tag{27o}$$

$$h_{t,d}^{\mathrm{last}} - h_{t,d}^{\mathrm{first}} + 1 - \sum_{b} p_{t,b,d} = w_{t,d} \qquad \forall t, d \tag{27p}$$

$$\sum_{j} vt_{t,d,j} \cdot \mathrm{ord}(j) = w_{t,d} \qquad \forall t, d \tag{27q}$$

$$\sum_{j} vt_{t,d,j} = 1 \qquad \forall t, d \tag{27r}$$

$$\sum_{g} V_{b,d} \cdot x_{s,g,b} \leq u_{s,d} \qquad \forall s, b, d \tag{27s}$$

$$\sum_{s,g} V_{b,d} \cdot P_{t,g} \cdot x_{s,g,b} \leq u_{t,d} \qquad \forall t, b, d \tag{27t}$$

$$\sum_{d} u_{s,d} - 1 \leq v_s \qquad \forall s \tag{27u}$$

$$\sum_{d} u_{t,d} - 1 \leq v_t \qquad \forall t \tag{27v}$$

$$x_{s,g,b} \in \{0, 1\}, \; y_{t,b,i} \in \{0, 1\} \tag{27w}$$

$$w_{t,d} \in \mathbb{N}, \; z_{s,d} \in \mathbb{N} \tag{27x}$$

$$vt_{t,d,j} \in \{0, 1\}, \; vs_{s,d,j} \in \{0, 1\} \tag{27y}$$

$$f_{s,d}^{\mathrm{first}} \in \mathbb{N}, \; f_{s,d}^{\mathrm{last}} \in \mathbb{N}, \; h_{t,d}^{\mathrm{first}} \in \mathbb{N}, \; h_{t,d}^{\mathrm{last}} \in \mathbb{N} \tag{27z}$$

$$p_{t,b,d} \in \{0, 1\}, \; n_{s,b,d} \in \{0, 1\} \tag{27aa}$$

$$u_{s,d} \in \{0, 1\}, \; u_{t,d} \in \{0, 1\} \tag{27ab}$$

$$v_s \in \mathbb{N}, \; v_t \in \mathbb{N} \tag{27ac}$$

3.6 Complexity

In the following a proof of $\mathcal{NP}$-hardness is given by showing that a well known NP-hard problem, the Graph Coloring Problem (GCP), is polynomially reducible to CTP.

An arbitrary instance of GCP consists of a graph $G = (V, E)$ and a number of colors k. The decision-version of the GCP asks the following: Does graph G admit a proper vertex coloring with k colors, such that no adjacent vertices take the same color?

To answer this question we solve a CTP with parameters $\delta_t = \delta_s = \gamma_{s,j} = \beta_{t,j} = \omega = \zeta_s = \zeta_t = HS = 0, \alpha_{s,g,b} = 1, C_{s,b} = 1$. This makes all constraints redundant, except for (27b), (27c) and (27d). Further assuming every student has exactly one request, $\sum_g R_{s,g} = 1\,\forall s$, makes constraint (27c) redundant.

For each vertex $v \in V$ in graph G, create a student s and a teachergroup g, and let the meeting request (s, g) represent vertex v. The set of vertices is hence represented by a setting of parameter $R_{s,g}$. If vertex $v_1 = (s_1, g_1)$ and vertex $v_2 = (s_2, g_2)$ are adjacent in graph G, create a teacher t and assign it to both g_1 and g_2, i.e. $P_{g_1,t} = P_{g_2,t} = 1$. I.e. every teacher will have exactly two meeting requests. Let the set of time slots B represent the set of colors (such that $|B| = k$).

Hence the GCP-instance is represented by the following CTP instance:

$$\max \quad \sum_{s,g,b} x_{s,g,b} \tag{28a}$$

$$\text{s.t.} \quad \sum_{b} x_{s,g,b} \leq R_{s,g} \qquad \forall s, g \tag{28b}$$

$$\sum_{s,g} P_{g,t} \cdot x_{s,g,b} \leq 1 \qquad \forall t, b \tag{28c}$$

$$x_{s,g,b} \in \{0, 1\} \tag{28d}$$

Constraint (28b) specifies that each vertex (meeting request) can at most be assigned one color (time slot). Constraint (28c) specifies that no teacher can be assigned more than one meeting in each time slot, which specifies that no adjacent vertices can take the same color.

To answer the question whether G is k-colorable, solve the CTP instance (28) and check if all meeting requests are assigned a time slot, i.e. $\sum_b x_{s,g,b} = 1 \quad \forall s, g, R_{s,g} = 1$. If so the answer is yes, otherwise the answer is no. Hence the Graph Coloring Problem is polynomially reducible to CTP, and CTP is therefore $\mathcal{NP}$-hard.

3.7 Defining weights

In the following the weights of the model are selected due to the preferences of the Danish high schools. MaCom A/S has greatly assisted this process. Table 1 lists all the weights in the model and their priority.

From analysis of previous consultations in the Danish high schools, it is noticed that the students rarely request more than five teacher groups for consultations. And even though the students have the opportunity to request more than five, they seldom use this option. From this analysis it is chosen to stick the request with priority higher

Table 1 Weight prioritizing

Weight	Symbol	Priority	Value dependency
(Max) Request fulfilling	$\alpha_{s,g,b}$	Very High	Priority of request (s, g) in time slot b
(Min) Teacher holes	$\beta_{t,j}$	High	Amount of requests of teacher t
(Min) Teacher sequence violation	ω	High	N/A
(Min) Teacher activities interruption	δ_t	Medium	N/A
(Min) Teacher multiple days	ζ_t	Medium	N/A
(Min) Student holes	$\gamma_{s,j}$	High	Amount of requests of student s
(Min) Student activities interruption	δ_s	Medium	N/A
(Min) Student multiple days	ζ_s	Medium	N/A

than five to the same weight. This gives the following function for the request weights $\alpha_{s,g,b}$:

$$\alpha_{s,g,b} = \begin{cases} \kappa_b + 12 - 2 \cdot (i-1) & i \leq 5 \\ \kappa_b + 2 & i \geq 6 \end{cases} \tag{29}$$

where $i \in \mathbb{Z}^+$ is the priority of request (g, s). Furthermore there is a set-point for each day given by b_d^* for which it is desired that the schedule plan for the given day is centered around. Let κ_b denote the penalty for assigning a request to time slot b, defined by

$$\kappa_b = -\sum_d \frac{V_{b,d} \cdot |b_d^* - b|}{|B_d|} \tag{30}$$

The cost of an undesirable break for a teacher, $\beta_{t,j}$, is defined as follows,

$$\beta_{t,j} = \begin{cases} 0 & j = 0 \\ j & j \geq 1 \wedge SCTP \\ j^{1+\frac{1.5}{\eta_t}} & j \geq 1 \wedge PCTP \end{cases} \tag{31}$$

where η_t is the number of requests for teacher groups where teacher t is a group member, $\eta_t = \sum_{s,g} P_{g,t} \cdot R_{s,g}$. I.e. $\beta_{t,j}$ depends on the number of requests for the given teacher t. The distribution of $\beta_{t,j}$ is chosen such that a teacher which few students have requested is given a high penalty for undesirable breaks. Likewise, a teacher with many requests has a low penalty for undesirable breaks. This is due to the fact that teachers with many requests will most likely have a more dense schedule, and are therefore not too picky about additional breaks. The reason why there is a difference between the weights for the different consultations types is due to the consultation interval. For the PCTP the consultation meetings are normally located in the evening, and hence we want to penalize the undesirable breaks. The SCTP is typically taken place in the daytime, i.e. the teachers are already at the high school, hence undesirable breaks are not that significant. The weight of an undesirable break for a student $\gamma_{s,j}$ is analogues,

$$\gamma_{s,j} = \begin{cases} 0 & j = 0 \\ j^{1+\frac{2}{\eta_s}} & j \geq 1 \end{cases} \tag{32}$$

where η_s is the number of requests of student s.

The cost for violating the length of a sequence for a teacher is given by ω.

$$\omega = 2 \tag{33}$$

In our model of the CTP the high school administration selects if interrupting other activities of the students or teachers is allowed. If this is not the case, the costs δ_t and δ_s are selected as infinitely high (implementation-wise the corresponding constraints are treated as hard-constraints). If interrupting other activities are not allowed, these costs are selected as a constant value,

$$\delta_t = \delta_s = \begin{cases} \infty & \text{Interrupting activities not allowed} \\ 4 & \text{Interrupting activities allowed} \wedge PCTP \\ 1 & \text{Interrupting activities allowed} \wedge SCTP \end{cases} \tag{34}$$

Like for the undesirable break cost $\beta_{t,j}$, we distinguish between the two consultations types. For the PCTP it is expensive to interrupt other activities since it is held in the evening and hence other activities are typically other types of meetings. The SCTP is held in the daytime, and it is allowed to ‘lent’ a student from a lecture for a small cost. The penalties for assigning students and teachers to multiple days are given by

$$\zeta_s = 8$$
$$\zeta_t = 1$$

4 Adaptive Large Neighborhood Search (ALNS)

In this section a heuristic alternative to solving the IP-model (27) is described. The performance of these two methods are compared in Sect. 6.

As the local search algorithm we have chosen to use the *Large Neighborhood Search* (LNS) proposed by Shaw (1998). Most local search algorithms explicitly defines the neighborhood, but the neighborhood in LNS is defined implicitly by a *destroy* and a *repair* method. The neighborhood of a solution is then defined as the set of solutions that can be reached by first applying a destroy method and then a repair method. In this article we will use ALNS, in which the LNS is extended by multiple destroy and repair methods. ALNS was first described in Pisinger and Ropke (2005), and has since been used with success on various problems, especially variants of Vehicle Routing Problems (VRP), see e.g. Ropke and Pisinger (2006), Laporte et al. (2010), Azi et al. (2010), Ribeiro and Laporte (2012), and Lei et al. (2011). A pseudo-code for the ALNS heuristic is shown in Algorithm 1.

Algorithm 1: Adaptive Large Neighborhood Search

```
   Input: a feasible solution $x^s_{g,b}$
1  solution $x_{best} = x; \pi = (1, \ldots, 1)$
2  repeat
3      $x' = x$
4      select destroy and repair methods $d \in \Omega^-$ and $r \in \Omega^+$ using $\pi$
5      select $q \in \mathbb{N}$
6      remove $q$ requests from $x'$ using $d$
7      reinsert removed requests into $x'$ using $r$
8      if $c(x') > c(x_{best})$ then
9          $x_{best} = x'$
10     if accept($x', x$) then
11         $x = x'$
12     update $\pi$
13 until stop-criterion met
14 return $x_{best}$
```

The sets of destroy and repair methods are denoted Ω^- and Ω^+, respectively. The variable π, which stores the weight of all destroy and repair methods, is introduced in line 1. Initially all methods have the same weight. In line 4 the weight parameter π is used to select the destroy and repair methods. In line 10 an accept function evaluates if the new solution should become the new current solution. The accept function can be implemented in different ways. We have chosen to implement a Simulated Annealing-like acceptance criterion, which will be described later.

An ALNS framework has the advantage of using different neighborhoods, such that the algorithm hopefully explores a large part of the solution space. For more information regarding ALNS we recommend Ropke and Pisinger (2006) and Pisinger and Ropke (2010).

4.1 ALNS scoring scheme

A central part of the ALNS algorithm is the scoring scheme of destroy and repair methods. A scoring scheme can essentially be characterized by two central topics; (1) How to quantify the performance of each heuristic. (2) The reaction factor, i.e. how sensitive is the selection process to recent records of performance.

We adapt a scoring scheme based on the technical report of Muller and Spoorendonk (2010), where performance is tracked by the percentage-wise gap between the new found solution and the current solution. This scoring scheme has the advantage of having few parameters to tune, and using the gap between solutions seems as a intuitively good way of measuring heuristic performance. Below the scoring scheme is explained in details.

Runs of the algorithm is divided into segments $\{t_0, t_1, \ldots, t_n\}$ each consisting of N_{it} iterations. Let π^t_i be the weight of heuristic i in segment t. The probability of choosing heuristic i in segment t is $\frac{\pi_i^t}{\sum_j \pi_j^t}$. At the end of each segment t, the following update is performed for all heuristics,

$$\pi_i^{t+1} = \rho \frac{\bar{\pi}_i^t}{a_i^t} + (1 - \rho)\pi_i^t \tag{35}$$

where a_i^t is the number of times heuristic i has been selected in segment t. $\bar{\pi}_i^t$ is the *observed* weight of heuristic i in segment t, which in each iteration is incremented depending on the quality of the new found solution. $\rho \in [0, 1]$ is the reaction factor. A high reaction factor means that the weights of a segment will be very dependent upon the observed weights of the previous segment.

The observed weight $\bar{\pi}_i^t$ is updated in each iteration. Let x be the current solution, and x' the new found solution by applying neighborhood i. In the technical report Muller and Spoorendonk (2010) the following formula is used,

$$\text{gap} = \frac{c(x') - c(x)}{c(x)} \tag{36}$$

$$\bar{\pi}_i^t = \bar{\pi}_i^t + m^{\text{gap}} \tag{37}$$

where m is a constant. We will use a slightly changed version of this formula, since we have observed that the gap formulated by (36) most often yields values of magnitude $\pm 10^{-4}$, meaning that the observed weight $\bar{\pi}_i^t$ will rarely change value of significant magnitude. Therefore we introduce a scale parameter in the formula,

$$\bar{\pi}_i^t = \bar{\pi}_i^t + m^{\min(\sigma \cdot \text{gap}, 1)} \tag{38}$$

where $\sigma \in \mathbb{R}^+$ is a parameter that needs tuning. We fix $m = 5$ and rely on the parameter tuning to set a suitable value for σ. The min-operator in the exponent of m is necessary to ensure the weight stay within a reasonable interval, in case we hit an iteration where the scaled gap is big and positive.

4.2 Request removal

The ALNS heuristic for the CTP makes use of two different removal heuristics, each searching a given removal neighborhood. The heuristics takes as input a given solution $x_{s,g,b}$ and an integer $q \in \mathbb{N}$. The output of the heuristics is the solution where q meetings have been removed. The value of q is selected as a random number which satisfies,

$$3 \leq q \leq \max\left(\xi \cdot \sum_{g,s} R_{s,g}, 5\right) \tag{39}$$

where $\xi \in]0, 1]$ is the maximum percentage of requests to remove. In accordance with Muller (2009) we decay ξ over time, starting with a high value ξ_{start} and ending with a smaller ξ_{end}. Given the runtime of the algorithm, we divide it into 100 segments such that ξ is decreased by $\frac{\xi_{\text{start}} - \xi_{\text{end}}}{100}$ in each segment. This decay of ξ means that the size of the searched neighborhood is progressively reduced. This has the advantage of only performing small changes towards the end of the solution process, where we expect a good solution has been found.

4.2.1 Random removal

The simplest removal heuristic, which randomly removes q meetings from the solution. This simple heuristic obviously has the effect of diversifying the search.

4.2.2 Shaw removal

This removal heuristic was introduced in Shaw (1997, 1998) where it is used on the VRP. In this section the heuristic is modified to suit the CTP. The general idea of the heuristic is to remove meetings which are somehow related, since there is a good chance that such requests can swap positions and possibly improve the solution. In this paper two factors determine if meeting i is related to meeting j: Similarity between students, and similarity between teachers. S_i and T_i indicates the set of students and teachers of meeting i, respectively. Notice that a meeting always contains exactly one student, i.e. $|S_i| = 1$. Let the measure of relatedness between meeting i and j be defined by $\mathcal{M}(i, j) \in [0, 1]$,

$$\mathcal{M}(i, j) = \frac{\left|T_i \cap T_j\right| + \left|S_i \cap S_j\right|}{\min(|T_i|, |T_j|) + 1} \tag{40}$$

I.e. relatedness is the percentage of individuals which is shared between the meetings, such that a high value of $\mathcal{M}$ means that the meetings are very related. This simple formulation of relatedness is done without any additional parameters. An alternative natural formulation would be to scale the student-relatedness and teacher-relatedness by two independent parameters. However we have chosen the shown formula due to its simplicity. An addition to the formula could be to introduce a term which determines time slot relatedness, although it should be noted that relatedness between slots is only directly relevant if there also exists some relatedness between students or teachers.

A pseudo code for Shaw removal is shown in Algorithm 2.

Algorithm 2: Shaw removal

```
Input: A feasible solution $x_{s,g,b}$, $q \in \mathbb{N}$, $p_{\text{shaw}} \in \mathbb{R}^+$
request: $r$ = a randomly selected meeting from $x_{s,g,b}$
set of requests : $D = \{r\}$
while $|D| < q$ do
    $r$ = randomly selected meeting from $D$
    $L$ = all meetings from $x_{s,g,b}$ not in $D$, sorted by decreasing similarity to $r$
    choose a random number $y^{p_{\text{shaw}}} \in [0, 1[$
    $l$ = element number $y^{p_{\text{shaw}}} \cdot |L|$
    $D = D \cup L[l]$
remove the meetings in $D$ from $x_{s,g,b}$
```

To avoid the situation where the same meetings are removed over and over, the algorithm is randomized. The level of randomness is controlled by the parameter $p_{\text{shaw}} \in \mathbb{R}^+$, $p_{\text{shaw}} \geq 1$. This means that p_{shaw} somehow defines how random the element is chosen, where $p_{\text{shaw}} = 1$ corresponds to completely random.

4.3 Repair heuristic

The repair heuristics are given a set of consultation meetings and a set of not granted meeting-requests.

4.3.1 Basic greedy heuristic

A trivial algorithm for the CTP is a simple greedy algorithm which places one request at a time in order of contribution to the objective. In each iteration of the algorithm this process is repeated until no more requests which improves the solution can be inserted. Implementation wise the algorithm suffers from cost-dependencies, since the contribution of inserting each request possibly changes after each insertion. This is slightly optimized by only recalculating the cost of those requests which the last insertion can possibly effect. I.e. recalculate the cost of those requests which has the same student as the last insertion, or if the teacher group overlaps with the one of the last insertion. This repair heuristic is used to create an initial feasible solution for the CTP.

4.3.2 Regret heuristics

The regret heuristic improves the basic greedy by incorporating a kind of look-ahead information when selecting a request to insert. Informally speaking, the heuristic aims at inserting the request which we will regret most if not inserted immediately. The regret heuristic has been used by Potvin and Rousseau (1993) and Pisinger and Ropke (2005) for the *VRP with Time Windows*. Let c_r^k denote the change in the objective value by inserting request r into the k^{th} best position. E.g. c_r^2 denotes the change in the objective value by inserting request r in the second best position. A Regret-2 heuristic will in each iteration choose to insert the request r where the difference between best and second best position is largest, i.e.

$$r := \underset{r \in R_g^s, c_r^1 > 0}{\arg\max} \left(c_r^1 - c_r^2\right) \tag{41}$$

The request r is inserted at its best position, so we restrict the heuristic to only look at requests where the best position is actually feasible and yields a positive change in objective. This restriction is necessary since the objective of the CTP contains both a minimization and a maximization part, and we are not interested in inserting requests which have negative impact on the objective. The heuristic can be extended by looking at k positions for each request. The request to insert is then chosen according to

$$r := \underset{r \in R_g^s, c_r^1 > 0}{\arg\max} \sum_{h=2}^{k} \left(c_r^1 - c_r^h \right) \tag{42}$$

We will in this paper incorporate the regret heuristic for several choices of k. The basic greedy algorithm from the previous section is a Regret-1 heuristic due to the tie-breaking rules. For a Regret-1 heuristic the most profitable request is inserted in each iteration. Most papers distinguish between Regret-1 and other regret heuristic, however implementation wise they are not very different. Setting $k = |B|$ corresponds to the full Regret-k heuristic.

Even though the regret heuristic is designed for VRP, it seems well suited for the CTP due to its assignment character. It seems valuable to attempt to predict which request is the most critical to insert. By some basic tests, we have chosen to use Regret-2, Regret-3, and Regret-$|B|$ as insertion heuristics.

4.4 Algorithm setup

According to Ropke and Pisinger (2006), using myopic repair heuristics, like those of this paper, one may apply noise to the objective function to obtain a more efficient algorithm. By applying noise, the repair heuristic will not always make the move that seems best locally. Ropke and Pisinger (2006) support this by strong computational results. However, preliminary tests show that, in our case, adding noise does not yield a more efficient algorithm. More precisely, noise was added such that it was controlled by a linear-scale parameter, and excessive tuning on this parameter yielded no convergence at all. I.e. this parameter had (close to) no impact on the algorithm efficiency. A similar result for the *Cumulative Capacitated VRP* is reported in Ribeiro and Laporte (2012).

In occurrence with Ropke and Pisinger (2006) we borrow an acceptance criteria from Simulated Annealing (SA). A solution x is always accepted if $c(x) > c(x_{\text{best}})$. If $c(x) < c(x_{\text{best}})$ then x is accepted with probability $\exp\left(-\frac{c(x_{\text{best}})-c(x)}{T}\right)$. In each iteration the temperature T is updated by $T = d_{\text{SA}}T$, where $0 < d_{\text{SA}} < 1$. Giving the *temperature control parameter* w_{SA}, $0 < w_{\text{SA}} < 1$, T is initially selected such that a solution is accepted with probability $\frac{1}{2}$ if its change in objective is w_{SA} percent worse than the initial solution x_0, i.e.

$$\exp\left(-\frac{(c(x_0) - (1 - w_{\text{SA}}) \cdot c(x_0))}{T_0}\right) = \tfrac{1}{2} \quad \Rightarrow \quad T_0 = \frac{w_{\text{SA}} \cdot c(x_0)}{\ln(2)} \tag{43}$$

This has the advantage of better adapting the temperature to each dataset.

Furthermore at the start of each segment (those of the ALNS scoring scheme), the current solution is set to the current best.

5 Parameter tuning

The proposed heuristic contains many free parameters. It is essential that these parameters are tuned to achieve good performance, see e.g. Adenso-Diaz and Laguna (2006) and Diao et al. (2003). Tuning of metaheuristics is usually done by rules-of-thumb and the researchers personal experience. However some well performing automated algorithms have lately been introduced, mainly *ParamILS* (Hutter et al. 2009) and *Race*-algorithms (Birattari 2005). In this paper we will use the *F-Race* algorithm for tuning, as its implementation burden seems light, and it has proven competitive for some heuristic methods, see Montero et al. (2010) and Pellegrini et al. (2010).

The main idea of a race algorithm is to sequentially process a set of data instances using all possible parameter configurations. In each iteration, the parameter configurations which are statistically inferior are eliminated. The algorithm is ran until one parameter configuration remains or the specified time limit is exceeded, see Algorithm 3. If more than one parameter configuration remains once the algorithm terminates, the one which in average has performed best is selected. The advantage of a race algorithm is that bad parameter configurations are eliminated early, such that no more valuable computation time is spend on evaluating these. The racing algorithm differs from most other tuning approaches in the sense that it only performs one algorithm run per parameter configuration per data instances. This relies on the proof in Birattari (2005) where it is shown that this is the optimal experimental setting in terms of variance of the estimated performance.

Algorithm 3: Race Tuning

```
Input:
$\Theta$: Set of parameter configurations
$T_{\text{exp}}$: Computation time of each experiment
$T_{\text{total}}$: Time limit
$\alpha$: Confidence level
$i = 0, S_\theta = \Theta, C_\theta = \emptyset$
while $t < T_{total}$ AND $|S_\theta| > 1$ do
    dataset = RandomSampled()
    foreach $\theta \in S_\theta$ do
        $C_\theta^i$ = EvalSolution($T_{\text{exp}}$, $\theta$, dataset)
    $i = i + 1$
    Drop inferior parameter configurations from $S_\theta$ by statistical test, using
    confidence level $\alpha$
```

In a F-Race algorithm, the *Friedman Two-way Analysis of Variance by Ranks* test is used to determine whether there is sufficient statistically evidence to eliminate parameter configurations from future iterations. If this is the case, then post-tests are performed where pairwise comparison between the best candidate and the remaining determines which configurations should be eliminated, if any. The F-Race algorithm has been successfully used for tuning in a number of cases, see e.g. Chiarandini et al. (2006), Becker et al. (2006) and Pellegrini and Birattari (2007).

Table 2 Final values of tuned parameters, found by the F-Race algorithm with confidence level $\alpha = 0.05$

Parameter	w_{SA}	d_{SA}	N_{it}	ρ	σ	ξ_{start}	ξ_{end}	p_{shaw}
Value	0.01	0.99	100	0.50	1000	0.30	0.0033	20

A problem of the described Racing algorithm, which applies to most tuning frameworks, is the so-called full-factorial design, meaning that the full set of parameter configuration is initially considered. This results in the F-Race becoming impractical and computational prohibitive, if there exists a large number of parameters and each parameters can take a modest number of values. This has been addressed in Balaprakash et al. (2007) by defining a probabilistic model on the set of all possible parameter configurations, such that a small set of parameter configurations is generated in each iteration of the tuning process. Elite configurations are used to update the model to bias the search around high quality parameter configurations. This version of F-Race is denoted *Iterative F-Race* (I/F-Race).

In this paper we use a simplified I/F-Race algorithm, where we start out with a small subset of parameter configurations, and based on the Race-results of these we manually construct new configurations, which are believed to be superior. One could think of this approach as a sort of manual iterative F-Race. Table 2 shows the best found parameter configuration. It should be mentioned that we set $b_{\text{SP}} = b_0$ and $\delta^s = \delta^t = \infty$, since these are the most common values chosen by the users. The datasets used are of the school year 2011/2012. w_{SA} is the temperature control parameter and d_{SA} is the decay parameter for the SA based acceptance criteria. N_{it} defines the number of iterations between resets. ρ and σ are reaction factor and the scale parameter for the ALNS scoring scheme, respectively. ξ_{start} and ξ_{end} are the destroy percentage in the beginning and in the end of the running time. Finally, p_{shaw} indicates how random the element is chosen in the Shaw removal.

6 Performance

The goal of this section is to evaluate the performance of the developed solution methods, the ALNS algorithm and solving the IP model. Also a comparison with the existing heuristic of Lectio is made. All tests are performed using nUnit 2.6 in C# 4.0 on a machine with an Intel i7-930@2.8 GHz CPU and 12 GB of RAM. No parallelization has been implemented.

6.1 Performance comparison between ALNS and Gurobi

In the following, the performance of the state-of-the-art MIP solver Gurobi 5.01 (currently top-ranked in the MIP benchmark of Mittelman 2012) and the implemented ALNS algorithm are compared. For both the PCTP and the SCTP, 100 datasets from the school-year 2011/2012 are selected from the database of Lectio.

In this experimental setup, the ALNS algorithm is run for 2 min. This low running time is due to the following: (1) The schools does generally not expect an algorithm to run longer, as they are usually not aware that it is a hard problem to solve. Some even believe the problem is trivial. (2) The ALNS "tailors-off" after a while, i.e only minor improvements are seen on the best found solution after the 2-min mark.

The Gurobi solver is run for 1 h, because we do not only want to evaluate the performance in terms of best found IP solution, we also want a good upper bound for the instances.

In the performance tests it is not allowed to interrupt other activities for the PCTP, i.e. $\delta_t = \delta_s = \infty$. For the SCTP interrupting activities is allowed. This is due to the fact that PCTP is normally arranged in the evening while SCTP is during the normal work-hours.

From Table 3 it is seen that ALNS in average finds solutions 4 % from optimum. Even though ALNS has far lower running time than Gurobi, it finds better solutions in almost all cases.

Table 4 shows the performance for the SCTP.

From Table 4 it is seen that ALNS in average finds solutions 4.1 % from optimum for the SCTP. This is lower than the average gap for Gurobi, which is 7.7 %.

From Table 3 it is seen that ALNS outperforms Gurobi for the PCTP. For the SCTP, the results are more blurred, but the ALNS still performs best in 70 out of 100 cases. What can also be seen from Tables 3 and 4 is that the standard deviation for the ALNS is low in all cases, and the maximum gap obtained across all datasets is considerably lower than the maximum gap which Gurobi obtains (even given the higher running time of Gurobi). This is important as the customers of Lectio expects a consistent and stable solution procedure.

6.2 Performance comparison of ALNS and current heuristic of Lectio

The current algorithm in Lectio is an undocumented heuristic, which initially attempts to fulfill every meeting request by assigning them to random time slots, and then attempts to find improving solutions with a hill-climber embedded in a SA framework. This heuristic does not support the SCTP. In this section, we compare the existing heuristic of Lectio with the implemented ALNS algorithm.

The comparison of algorithms for the PCTP is done by adapting the objective of the ALNS so it matches the one of the implemented SA algorithm, which yields the following changes:

- Since the SA algorithm attempts to fulfill all meeting requests, we set α^s_g to a huge value for all meeting requests, effectively making the ALNS behave the same way.
- We set $\beta^t = \gamma^s = 2$.
- The SA algorithm does not allow interrupting of activities, i.e. $\delta^t = \delta^s = \infty$.
- The time slot set-point setting of the SA solver is broken, so we set $\kappa = 0$ and likewise for the SA solver.
- The violation of sequence length for teachers is penalized in quadratic way. This means that term (21) is now written as $-\sum_{t,b,d} \left(y_{t,b,d}\right)^{\omega}$, and $\omega = 2$.

Table 3 Gurobi 5.01 and ALNS for the PCTP on 100 datasets

					Gurobi 5.01			ALNS		
	$\lvert B\rvert$	$\sum R$	$\frac{\sum R}{\lvert S\rvert}$	$\frac{\sum R}{\lvert T\rvert}$	x	UB	Gap (%)	$\bar{x}$	σ	Gap (%)
Alleroed	12	51	3.0	2.4	**485.0**	485.0	0.0	484.8	0.1	0.0
Alssund	18	84	2.5	4.0	**850.6**	850.6	0.0	849.4	0.6	0.1
Aurehoej1	18	537	4.0	9.9	3,270.3	3,774.3	15.4	**3,655.9**	6.9	3.2
Aurehoej2	18	409	3.8	6.6	3,033.1	3,300.5	8.8	**3,219.5**	3.9	2.5
Broenderslev	24	241	4.0	4.8	1,656.2	1,965.1	18.7	**1,905.6**	3.0	3.1
CPHWEST	39	133	3.7	5.8	990.5	1,124.7	13.5	**1,044.5**	3.0	7.7
DetKristne	32	247	4.1	11.2	1,553.3	1,978.5	27.4	**1,830.9**	4.9	8.1
Dronninglund1	30	108	4.9	7.7	726.0	801.5	10.4	**782.0**	4.1	2.5
Dronninglund2	30	94	4.1	8.6	636.6	672.7	5.7	**664.5**	0.6	1.2
Egedal	27	408	3.4	6.3	3,143.5	3,625.0	15.3	**3,558.2**	3.7	1.9
Egaa	24	265	3.2	9.8	2,101.7	2,374.0	13.0	**2,318.5**	7.7	2.4
Esbjerg1	24	345	3.9	9.6	2,306.2	2,465.3	6.9	**2,402.1**	4.2	2.6
Esbjerg2	24	307	3.5	4.3	2,117.6	2,439.9	15.2	**2,314.3**	2.4	5.4
Esbjerg3	24	255	3.8	9.4	1,770.0	1,888.1	6.7	**1,839.5**	2.8	2.6
Esbjerg4	24	351	4.0	9.2	2,493.8	2,700.5	8.3	**2,612.3**	4.5	3.4
Frederikssund	24	49	3.3	4.5	**404.2**	406.8	0.7	403.8	0.4	0.8
Frederiksvaerk	8	74	2.4	3.2	**699.0**	699.0	0.0	697.9	0.6	0.2
Gefion	18	479	3.1	7.2	3,316.6	4,248.9	28.1	**3,958.1**	23.7	7.4
Gladsaxe	40	901	4.1	11.0	5,516.0	7,163.7	29.9	**6,950.7**	15.1	3.1
Greve	18	336	4.5	4.4	2,133.1	2,535.9	18.9	**2,482.6**	4.4	2.2
Haslev1	18	123	2.9	5.4	1,051.5	1,077.1	2.4	**1,060.2**	0.5	1.6
Haslev2	18	122	3.2	4.2	983.5	1,019.6	3.7	**988.7**	2.6	3.1
Herlufsholm1	24	143	4.3	14.3	853.1	918.9	7.7	**894.5**	1.2	2.7
Herlufsholm2	24	88	4.9	6.8	599.5	671.3	12.0	**621.8**	1.9	8.0
Herning1	27	118	3.8	3.6	0.0	0.0	0.0	0.0	0.0	0.0
Herning2	27	75	3.4	6.3	0.0	0.0	0.0	0.0	0.0	0.0
Herning3	27	140	3.7	4.5	2.0	2.0	0.0	2.0	0.0	0.0
Himmelev	34	453	3.7	8.4	3,118.1	3,736.1	19.8	**3,471.7**	8.8	7.6
Hjoerring	30	179	3.8	2.1	751.2	1,229.9	63.7	**1,009.2**	15.3	21.9
HorsensGym	18	123	3.2	3.4	**1,131.5**	1,133.8	0.2	1,129.4	1.1	0.4
HorsensStats	21	143	2.9	3.0	1,217.8	1,289.3	5.9	**1,253.8**	1.9	2.8
Ikast-Brande	30	52	3.5	5.8	419.4	463.7	10.6	**449.7**	0.4	3.1
Johannesskolen1	24	165	4.5	7.9	1,131.2	1,296.8	14.6	**1,188.3**	4.3	9.1
Johannesskolen2	24	97	3.7	7.5	701.4	786.3	12.1	**743.9**	5.0	5.7
Johannesskolen3	28	135	3.6	7.5	519.2	581.4	12.0	**519.9**	3.2	11.8
Kalundborg	27	299	3.0	4.8	2,281.0	2,587.0	13.4	**2,458.3**	5.3	5.2
Kolding	18	80	3.5	4.0	**721.4**	725.2	0.5	721.0	0.2	0.6
Langkær1	18	52	3.5	2.6	**471.1**	471.1	0.0	470.6	0.3	0.1
Langkær2	18	90	3.5	3.8	788.9	814.6	3.3	**805.9**	0.5	1.1

Table 3 continued

					Gurobi 5.01			ALNS		
	$\lvert B \rvert$	$\sum R$	$\frac{\sum R}{\lvert S \rvert}$	$\frac{\sum R}{\lvert T \rvert}$	x	UB	Gap (%)	$\bar{x}$	σ	Gap (%)
Middelfart	27	223	3.1	6.2	1,682.5	1,923.6	14.3	**1,788.1**	7.4	7.6
Morsoe1	27	105	3.9	10.5	768.8	834.2	8.5	**804.2**	3.2	3.7
Morsoe2	27	113	4.7	10.3	732.9	815.2	11.2	**778.4**	4.2	4.7
Munkensdam1	21	256	3.6	4.7	1,871.9	2,251.6	20.3	**2,198.6**	5.3	2.4
Munkensdam2	21	345	3.4	6.3	2,352.8	2,931.6	24.6	**2,846.8**	6.7	3.0
NielsSteensens1	36	117	5.9	7.8	715.8	781.9	9.2	**757.2**	3.9	3.3
NielsSteensens2	30	328	7.5	17.3	1,249.9	1,764.9	41.2	**1,656.6**	5.9	6.5
NielsSteensens3	30	365	7.9	20.3	1,125.7	1,895.7	68.4	**1,800.3**	7.4	5.3
NielsSteensens4	30	234	6.7	13.8	1,101.6	1,230.0	11.7	**1,144.4**	5.2	7.5
NielsSteensens5	30	263	6.3	17.5	1,461.7	1,634.2	11.8	**1,557.1**	5.7	5.0
Noerre	18	422	2.7	7.8	3,574.1	4,031.6	12.8	**3,944.5**	5.2	2.2
Nordfyns	23	192	2.6	6.6	1,761.9	1,863.5	5.8	**1,795.5**	6.2	3.8
Nordsjaellands1	34	1,187	6.4	25.3	6,001.7	7,018.5	16.9	**6,597.3**	27.9	6.4
Nordsjaellands2	34	1,038	6.6	23.1	2,298.7	2,626.1	14.3	**2,453.2**	14.7	7.1
Nordsjaellands3	34	457	6.3	13.9	2,225.4	2,858.3	28.4	**2,634.7**	7.8	8.5
Nordsjaellands4	34	163	4.9	9.6	1,100.9	1,210.0	9.9	**1,172.4**	2.9	3.2
Nordsjaellands5	40	712	5.6	19.8	2,543.1	4,796.2	88.6	**4,460.5**	16.6	7.5
Nordsjaellands6	34	780	6.1	19.0	2,500.4	4,899.0	95.9	**4,612.2**	14.2	6.2
Nordsjaellands7	34	880	6.1	19.1	2,763.3	3,047.9	10.3	**2,894.5**	6.3	5.3
Nordsjaellands8	34	23	1.6	3.3	**242.1**	242.1	0.0	241.9	0.3	0.1
Nordsjaellands9	34	949	6.2	22.1	3,202.9	5,519.0	72.3	**5,037.1**	31.8	9.6
Nordsjaellands10	34	31	1.9	4.4	**272.2**	272.9	0.2	269.1	3.3	1.4
Nyborg	24	119	3.2	5.7	55.4	55.4	0.0	55.4	0.0	0.0
Nykoebing	24	182	3.0	3.2	1,447.1	1,502.2	3.8	**1483.1**	1.2	1.3
NZahles1	25	324	4.3	9.5	2,116.2	2,456.3	16.1	**2,365.7**	5.5	3.8
NZahles2	24	301	4.1	8.9	2,035.3	2,280.1	12.0	**2,217.8**	6.5	2.8
Odder	18	95	4.0	7.3	740.0	773.5	4.5	**762.7**	1.5	1.4
Odsherreds	21	193	3.4	5.0	1,533.4	1,619.6	5.6	**1,595.4**	2.2	1.5
Risskov1	15	65	3.1	4.6	**539.7**	539.7	0.0	536.7	2.5	0.6
Risskov2	15	149	3.5	5.7	**1,263.2**	1,273.1	0.8	1,256.9	2.9	1.3
Risskov3	15	181	3.7	6.2	**1,396.9**	1,406.8	0.7	1,389.7	4.8	1.2
Roedkilde	18	266	3.6	9.2	2,161.1	2,352.9	8.9	**2,325.2**	5.2	1.2
Roedovre	51	779	3.6	10.3	1,513.8	2,032.4	34.3	**1,661.7**	20.1	22.3
Rosborg1	24	218	3.5	9.9	1,742.7	1,876.3	7.7	**1,827.9**	5.4	2.7
Rosborg2	28	268	3.7	11.7	1,960.9	2,297.2	17.2	**2,223.3**	11.4	3.3
Rosborg3	28	487	1.9	9.6	4,568.5	4,939.6	8.1	**4,750.0**	10.1	4.0
Rosborg4	26	235	3.6	8.4	1,713.8	2,033.2	18.6	**1,960.4**	11.1	3.7
Roskilde	48	263	3.7	6.4	1,716.4	2,251.7	31.2	**2,112.8**	6.4	6.6
Rybners	24	267	3.1	6.9	1,923.3	2,472.3	28.5	**2,402.5**	3.8	2.9

Table 3 continued

					Gurobi 5.01			ALNS		
	$\lvert B\rvert$	$\sum R$	$\frac{\sum R}{\lvert S\rvert}$	$\frac{\sum R}{\lvert T\rvert}$	x	UB	Gap (%)	$\bar{x}$	σ	Gap (%)
SanktAnnae	21	320	4.1	7.0	2,115.0	2,498.7	18.1	**2,381.6**	5.6	4.9
Skive	36	220	3.6	12.9	1,611.0	1,903.2	18.1	**1,850.3**	2.9	2.9
Slagelse	30	85	3.0	3.9	**805.3**	805.4	0.0	802.6	0.8	0.4
Solroed1	16	341	3.3	7.4	2,418.2	2,468.7	2.1	**2,436.8**	6.1	1.3
Solroed2	16	415	3.4	7.2	3,083.3	3,317.9	7.6	**3,263.0**	5.2	1.7
Soroe1	24	369	3.7	8.2	2,587.1	3,111.2	20.3	**2,947.3**	12.1	5.6
Soroe2	33	335	4.2	5.6	1,594.3	2,649.3	66.2	**2,255.0**	12.3	17.5
Stenhus	18	221	4.3	3.2	0.0	0.0	0.0	0.0	0.0	0.0
Stoevring	24	62	3.7	4.4	**521.4**	521.4	0.0	520.5	0.5	0.2
Struer1	30	237	3.3	4.4	1,596.8	1,801.2	12.8	**1,656.9**	2.1	8.7
Struer2	30	333	3.3	8.8	2,301.3	2,790.8	21.3	**2,534.0**	7.9	10.1
Svendborg1	18	96	2.1	2.9	**991.4**	991.4	0.0	991.1	0.2	0.0
Svendborg2	18	134	2.6	4.5	**1,289.0**	1,289.1	0.0	1,288.0	0.3	0.1
Taarnby	36	791	4.6	11.0	4,397.7	5,918.0	34.6	**5,609.2**	17.7	5.5
UCH	32	104	1.0	17.3	**949.8**	949.8	0.0	922.6	0.0	2.9
ViborgGym1	30	206	4.3	5.2	1,367.7	1,486.5	8.7	**1,434.0**	2.6	3.7
ViborgGym2	30	149	4.4	5.3	1,094.0	1,146.6	4.8	**1,133.4**	0.9	1.2
ViborgGym3	30	294	3.7	4.6	2,153.6	2,275.5	5.7	**2,211.7**	1.8	2.9
ViborgHandel	30	324	4.2	18.0	2,111.8	2,615.9	23.9	**2,526.7**	9.7	3.5
ViborgKatedral	40	337	4.8	11.2	1,935.5	2,516.5	30.0	**2,313.4**	8.6	8.8
Vordingborg1	16	315	3.8	6.3	2,222.0	2,358.8	6.2	**2,304.3**	2.8	2.4
Vordingborg2	16	239	3.3	5.6	1,867.5	1,950.0	4.4	**1,924.6**	2.0	1.3
Average	26	279	3.9	8.3			14.8			4.0
Max	51	1187	7.9	25.3			95.9			22.3

For each dataset is listed the number of time slots "$|B|$", the total number of meeting requests "$\sum R$", the average number of requests pr. student, and the average number of requests pr. teacher. For Gurobi is shown the final objective value "x", the best bound found "UB", and the reported gap between these two. For ALNS, column "$\bar{x}$" is the mean performance of the algorithm over 10 runs, and column "σ" is the standard deviation for these runs. Finally, column "Gap" is the gap of mean performance and the upper bound found by Gurobi. The best found solution is marked with bold for each instance

We evaluate the algorithms on 100 randomly selected datasets for the school-year 2009/2010. The reason a new batch of datasets is selected for this test is that the existing heuristic of Lectio does not support all features mentioned in this paper. Due to customer requests, Lectio is continuously developed, and this also effects the CTP. E.g. datasets from the school-year 2009/2010 does not support features such as multiple days for a consultation.

Experience has shown that the SA algorithm needs long runtime to provide meaningful solutions. We set runtime equal to 10 min, which is significantly higher than the preferable runtime of the high schools, as described in Sect. 6.1. Furthermore, to

Table 4 Gurobi and ALNS for the SCTP on 100 datasets

				Gurobi 5.01			ALNS		
	$\|B\|$	$\sum R$	$\frac{\sum R}{\|T\|}$	x	UB	Gap (%)	$\bar{x}$	σ	Gap (%)
Aabenraa	60	226	4.3	**2,461.2**	2,495.7	1.4	2,387.5	3.3	4.5
Broendby1	21	69	3.8	677.3	707.3	4.4	**683.2**	2.8	3.5
Broendby2	14	69	4.3	**770.2**	779.5	1.2	768.6	1.0	1.4
Broendby3	24	62	3.4	609.0	632.8	3.9	**614.8**	1.3	2.9
Broenderslev1	69	115	3.5	**1,335.4**	1,340.9	0.4	1,302.4	2.1	3.0
Broenderslev2	102	115	3.5	**1,253.4**	1,272.5	1.5	1,236.0	1.6	3.0
Christianshavns	43	210	4.8	**2,329.7**	2,423.7	4.0	2,223.7	4.4	9.0
Dronninglund1	100	134	4.2	1,442.7	1,481.3	2.7	**1,453.5**	2.1	1.9
Dronninglund2	60	134	4.3	**1,553.3**	1,561.9	0.6	1,537.5	2.8	1.6
Egaa	29	214	4.8	2,257.5	2,457.1	8.8	**2,376.2**	4.0	3.4
Falkoner1	30	64	3.4	**671.6**	679.0	1.1	668.0	2.6	1.6
Falkoner2	37	206	4.2	2,188.3	2,345.6	7.2	**2,266.1**	5.0	3.5
Falkoner3	30	64	3.4	664.9	672.3	1.1	664.9	0.0	1.1
Grenaa1	28	122	3.8	1,280.7	1,380.3	7.8	**1,325.4**	3.1	4.2
Grenaa2	24	122	3.8	1,249.2	1,330.7	6.5	**1,290.3**	3.8	3.1
Greve1	28	157	3.3	1,575.1	1,762.2	11.9	**1,693.3**	3.2	4.1
Greve2	62	259	4.1	2,805.4	3,035.8	8.2	**2,913.5**	7.4	4.2
Greve3	20	51	3.2	**566.4**	569.9	0.6	566.3	0.0	0.6
Gribskov1	74	182	4.1	**1,867.8**	1,914.6	2.5	1,787.6	3.2	7.1
Herlev1	24	71	2.8	729.9	751.5	3.0	**730.2**	0.9	2.9
Herlev2	29	78	2.8	682.1	791.7	16.1	**750.2**	1.2	5.5
Hoeng1	21	66	3.5	607.9	686.5	12.9	**621.5**	0.4	10.5
Hoeng2	23	98	3.9	1,029.8	1,071.8	4.1	**1,038.5**	2.2	3.2
Hoeng3	22	45	2.7	392.7	481.7	22.7	**408.3**	1.2	18.0
Hoeng4	23	56	2.6	589.1	612.4	4.0	**589.2**	1.8	3.9
Koebenhavns1	16	143	3.9	1,239.8	1,278.3	3.1	**1,242.0**	6.1	2.9
Koebenhavns2	16	100	3.7	**786.4**	786.5	0.0	785.6	0.8	0.1
Koebenhavns3	16	100	3.7	725.8	725.9	0.0	725.8	0.0	0.0
Koebenhavns4	25	146	3.8	1,402.3	1,486.1	6.0	**1,424.6**	0.8	4.3
Koege1	30	255	8.5	1,348.2	2,475.2	83.6	**2,333.2**	10.3	6.1
Koege2	36	261	6.2	**2,381.3**	2,493.7	4.7	2,045.7	5.2	21.9
Koege3	74	258	8.6	**2,775.1**	2,903.2	4.6	2,622.9	5.0	10.7
Kolding1	24	219	5.0	2,283.5	2,430.9	6.5	**2,348.7**	4.8	3.5
Kolding2	45	174	3.8	**1,934.9**	2,005.3	3.6	1,908.8	4.1	5.1
Langkaer1	62	215	5.4	2,195.8	2,472.6	12.6	**2,239.4**	7.8	10.4
Langkaer2	60	216	5.4	**2,351.0**	2,481.0	5.5	2,240.1	9.3	10.8
Langkaer3	60	216	5.4	**2,359.1**	2,472.8	4.8	2,258.3	6.6	9.5
Langkaer4	30	57	3.8	546.4	596.1	9.1	**566.2**	3.6	5.3
Langkaer5	56	217	5.6	**2,282.3**	2,493.5	9.3	2,253.3	3.4	10.7

Table 4 continued

				Gurobi 5.01			ALNS		
	$\|B\|$	$\sum R$	$\frac{\sum R}{\|T\|}$	x	UB	Gap (%)	$\bar{x}$	σ	Gap (%)
Langkaer6	62	56	3.7	**631.5**	652.6	3.3	629.2	0.7	3.7
Mariagerfjord1	29	123	4.0	1,227.0	1,387.2	13.1	**1,318.6**	2.0	5.2
Mariagerfjord2	29	123	4.0	1,269.0	1,402.6	10.5	**1,345.3**	4.0	4.3
Marselisborg1	22	102	3.4	1,021.3	1,087.4	6.5	**1,045.4**	1.9	4.0
Marselisborg2	17	106	3.3	**1,036.2**	1,046.9	1.0	1,035.4	1.0	1.1
Marselisborg3	22	105	3.9	1,049.3	1,156.3	10.2	**1,098.3**	4.3	5.3
Marselisborg4	17	96	3.2	**948.6**	955.2	0.7	947.2	0.9	0.8
Munkensdam	43	191	5.6	**2,179.9**	2,203.8	1.1	2,067.9	4.6	6.6
Nordfyns1	22	173	5.1	1,871.9	1,974.3	5.5	**1,926.6**	1.6	2.5
Nordfyns2	21	173	5.2	1,846.9	1,975.6	7.0	**1,929.9**	2.6	2.4
Nordfyns3	22	173	5.1	1,831.5	1,973.7	7.8	**1,908.1**	2.1	3.4
Nordfyns4	21	173	4.1	1,438.0	1,538.2	7.0	**1,478.3**	3.7	4.1
Noerresundby	31	303	4.7	2,959.5	3,437.4	16.2	**3,291.7**	2.6	4.4
NZahles1	13	69	3.3	615.7	636.9	3.5	**619.1**	0.9	2.9
NZahles2	13	62	3.9	**512.1**	524.3	2.4	509.5	1.3	2.9
Odsherreds	49	119	3.8	**1,365.1**	1,372.6	0.6	1,289.4	3.4	6.5
Oeregaard1	20	219	5.6	2,257.3	2,296.6	1.7	**2,258.1**	5.6	1.7
Oeregaard2	20	213	5.0	1,728.9	1,778.5	2.9	**1,743.9**	3.2	2.0
Oeregaard3	20	219	5.6	2,327.5	2,372.3	1.9	**2,340.2**	3.3	1.4
Oeregaard4	20	219	5.6	2,338.4	2,373.5	1.5	**2,339.9**	2.6	1.4
Risskov	36	215	6.1	**2,400.4**	2,427.4	1.1	2,353.2	2.0	3.2
Roedkilde	18	230	4.4	2,452.3	2,534.1	3.3	**2,495.7**	3.1	1.5
Rosborg1	22	257	4.9	2,756.7	2,895.6	5.0	**2,837.4**	3.2	2.1
Rosborg2	22	257	4.8	2,651.0	2,860.6	7.9	**2,805.4**	3.3	2.0
SanktAnnae1	23	149	3.6	1,554.1	1,675.1	7.8	**1,580.4**	3.9	6.0
SanktAnnae2	24	165	3.8	1,682.9	1,850.2	9.9	**1,753.4**	4.0	5.5
SanktAnnae3	17	21	2.6	197.9	197.9	0.0	197.9	0.0	0.0
SanktAnnae4	31	162	4.2	1,359.0	1,719.9	26.6	**1,598.9**	3.6	7.6
Skanderborg1	57	232	3.9	2,440.0	2,640.1	8.2	**2,547.4**	3.4	3.6
Skanderborg2	60	229	4.9	**2,369.5**	2,414.4	1.9	2,320.4	3.4	4.1
Skive1	16	140	3.3	1,417.5	1,458.6	2.9	**1,430.9**	3.8	1.9
Skive2	31	103	2.6	865.5	1,062.6	22.8	**995.5**	2.5	6.8
Skive3	31	140	3.3	1,189.7	1,452.9	22.1	**1,372.7**	4.4	5.8
Skive4	16	21	2.1	**227.8**	227.8	0.0	227.6	0.1	0.1
Skive5	16	98	3.0	**960.9**	972.9	1.3	960.7	0.9	1.3
Skive6	16	110	3.1	1,106.9	1,144.7	3.4	**1,119.3**	2.0	2.3
Skive7	31	134	3.0	1,152.9	1,365.6	18.5	**1,284.6**	3.2	6.3
Skive8	16	107	3.0	1,006.8	1,015.8	0.9	**1,007.1**	0.9	0.9
Skive9	31	100	3.0	907.8	1,034.9	14.0	**983.0**	4.6	5.3

Table 4 continued

				Gurobi 5.01			ALNS		
	$\|B\|$	$\sum R$	$\frac{\sum R}{\|T\|}$	x	UB	Gap (%)	$\bar{x}$	σ	Gap (%)
Soenderborg1	22	234	3.7	2,105.1	2,590.6	23.1	**2,475.3**	4.9	4.7
Soenderborg2	22	236	3.4	2,095.1	2,703.9	29.1	**2,577.9**	4.2	4.9
Soenderborg3	21	236	3.4	2,452.5	2,688.1	9.6	**2,597.3**	4.0	3.5
Soenderborg4	22	235	3.4	1,927.6	2,681.3	39.1	**2,554.2**	3.7	5.0
Solroed1	18	242	4.6	1,935.5	2,181.4	12.7	**2,130.3**	6.8	2.4
Solroed2	20	22	1.8	228.6	228.6	0.0	228.6	0.0	0.0
Solroed3	17	22	1.8	223.3	223.3	0.0	223.3	0.0	0.0
Solroed4	54	243	4.6	2,309.9	2,562.4	10.9	**2,354.2**	5.2	8.8
Solroed5	20	243	4.5	1,555.0	2,185.8	40.6	**2,054.9**	8.0	6.4
Solroed6	17	215	4.2	1,703.0	1,835.0	7.8	**1,775.5**	6.8	3.4
Solroed7	17	194	4.0	1,599.6	1,748.0	9.3	**1,679.2**	2.6	4.1
Vejen1	10	41	2.4	424.2	424.2	0.0	424.2	0.0	0.0
Vejen2	19	126	3.8	1,171.2	1,225.0	4.6	**1,198.5**	3.0	2.2
Vejen3	19	125	4.2	1,172.1	1,234.9	5.4	**1,204.6**	2.4	2.5
Vejen4	19	125	4.2	1,130.2	1,205.3	6.6	**1,172.5**	2.4	2.8
Viborg1	19	105	3.8	1,004.3	1,100.3	9.6	**1,034.2**	0.6	6.4
Viborg2	49	187	4.4	**2,081.7**	2,153.0	3.4	2,060.0	4.8	4.5
Viby1	20	124	3.7	**1,278.3**	1,279.4	0.1	1,256.9	1.2	1.8
Viby2	13	93	5.2	957.5	957.5	0.0	957.5	0.0	0.0
Viby3	8	45	2.7	480.0	480.0	0.0	480.0	0.0	0.0
Viby4	16	93	5.2	**1,057.7**	1057.7	0.0	1,053.3	0.3	0.4
Viby5	21	123	3.7	**1,378.6**	1,378.8	0.0	1,356.1	1.1	1.7
Average	30	148	4.1			7.7			4.1
Max	102	303	8.6			83.6			21.9

Columns are defined in analogous way to Table 3. The average number of requests pr. student is not shown, as it is 1.0 in all cases

reduce the influence of stochastic behavior, we perform 10 runs on each dataset with each solver.

Table 5 shows the average performance, the standard deviation and the number of unassigned requests of both algorithms. Recall that in this test both algorithms attempt to fulfill every meeting request. However it cannot be guaranteed that the algorithms can find a solution which satisfies this, nor that it even exists. Furthermore we compare the algorithms in the domain of the SA algorithm, and in this domain it is only attempted to minimize the different costs, i.e. no benefit is made from fulfilling meeting requests. This means it would not be fair to compare solutions which does not have the same amount of unassigned requests, since additional fulfilled requests will potentially yield additional cost of e.g. number of breaks, interrupted activities, etc. Therefore we enforce the following criterion to determine whether the comparison of solutions for a dataset is valid: The difference in the number of unassigned requests

Table 5 Comparison of performance of the SA algorithm and the ALNS algorithm

	Lectio SA			ALNS			
	$\bar{x}$	σ	#UA	$\bar{x}$	σ	#UA	Diff.
Aabenraa1	640.90	83.36	17.0	**555.00**	0.00	17.0	85.90
Aabenraa2	384.00	41.86	9.0	**315.90**	18.27	9.0	68.10
~~Aabenraa3~~	~~41.60~~	~~11.27~~	~~12.4~~	~~123.90~~	~~7.61~~	~~9.0~~	~~N/A~~
~~Aabenraa4~~	~~126.90~~	~~59.36~~	~~13.5~~	~~166.80~~	~~53.45~~	~~12.0~~	~~N/A~~
Aabenraa5	**36.00**	9.30	0.0	37.60	0.84	0.0	-1.60
Aabenraa6	0.00	0.00	60.0	0.00	0.00	60.0	0.00
Aabenraa7	0.00	0.00	52.0	0.00	0.00	52.0	0.00
Aurehoej	283.30	37.99	12.8	**165.70**	12.72	12.0	117.60
~~Birkeroed1~~	~~542.10~~	~~88.05~~	~~40.2~~	~~171.70~~	~~37.57~~	~~35.5~~	~~N/A~~
Birkeroed2	493.80	88.41	30.1	**92.40**	19.21	29.5	401.40
Bjerringbro	2994.60	390.30	22.0	**1393.20**	16.90	22.0	1601.40
~~Borupgaard1~~	~~36.60~~	~~11.66~~	~~10.9~~	~~176.70~~	~~11.98~~	~~2.0~~	~~N/A~~
~~Borupgaard2~~	~~16.00~~	~~8.03~~	~~51.2~~	~~221.40~~	~~1.71~~	~~41.0~~	~~N/A~~
~~Borupgaard3~~	~~32.80~~	~~23.67~~	~~26.3~~	~~115.80~~	~~10.34~~	~~20.0~~	~~N/A~~
Broenderslev	154.40	51.04	0.0	**22.20**	6.96	0.0	132.20
CPHWEST	1244.40	38.56	0.0	**1105.00**	0.00	0.0	139.40
Esbjerg	0.00	0.00	407.0	0.00	0.00	407.0	0.00
Fjerritslev	2088.50	72.25	0.0	**1904.20**	6.34	0.0	184.30
Frederikssund1	330.40	39.98	0.0	**257.70**	6.34	0.0	72.70
Frederikssund2	436.40	22.85	0.0	**340.00**	4.59	0.0	96.40
Frederiksvaerk	25.90	1.85	34.0	**25.00**	0.00	34.0	0.90
GlHellerup1	130.90	29.37	4.0	**29.50**	5.78	4.0	101.40
GlHellerup2	268.70	36.58	1.0	**88.60**	12.95	1.0	180.10
Gefion	112.40	22.04	0.0	**16.70**	3.65	0.0	95.70
~~Gladsaxe~~	~~85.40~~	~~16.40~~	~~87.8~~	~~1034.00~~	~~37.54~~	~~27.0~~	~~N/A~~
Haderslev1	0.00	0.00	225.0	0.00	0.00	225.0	0.00
Haderslev2	0.00	0.00	185.0	0.00	0.00	185.0	0.00
Haslev1	1239.10	56.96	0.0	**1181.00**	0.00	0.0	58.10
Haslev2	0.00	0.00	51.0	0.00	0.00	51.0	0.00
Hasseris1	10.20	4.52	0.0	**0.90**	0.57	0.0	9.30
Hasseris2	68.80	18.94	0.0	**14.90**	2.56	0.0	53.90
Hasseris3	34.30	25.95	2.9	**5.10**	2.02	3.0	29.20
Hasseris4	11.10	6.23	0.0	**0.10**	0.32	0.0	11.00
Herlufsholm1	46.70	8.90	0.0	**34.30**	5.54	0.0	12.40
Herlufsholm2	86.30	14.35	1.2	**60.20**	4.05	0.0	26.10
~~Herlufsholm3~~	~~2.40~~	~~2.41~~	~~38.4~~	~~0.00~~	~~0.00~~	~~37.0~~	~~N/A~~
~~Herlufsholm4~~	~~267.30~~	~~32.27~~	~~3.3~~	~~176.90~~	~~9.80~~	~~2.0~~	~~N/A~~
Herlufsholm5	15.70	11.86	0.0	**1.60**	1.84	0.0	14.10
Herning1	843.30	119.31	1.0	**632.10**	9.00	1.0	211.20
Herning2	103.50	16.25	0.0	**74.20**	0.42	0.0	29.30
Himmelev	262.10	35.44	0.0	**96.30**	16.22	0.0	165.80
Horsens	17.20	7.50	2.0	**6.50**	1.35	2.0	10.70
Kongsholm	102.20	39.82	21.4	**62.30**	8.64	21.0	39.90
~~Langkaer~~	~~29.20~~	~~8.22~~	~~11.7~~	~~190.90~~	~~10.38~~	~~0.0~~	~~N/A~~
Mariagerfjord	190.00	52.75	0.9	**89.00**	17.58	1.0	101.00
Morsoe1	17.20	9.74	3.0	**0.50**	0.71	3.0	16.70
Morsoe2	44.00	21.91	0.0	**2.70**	1.42	0.0	41.30
Mulernes1	21.50	8.22	0.0	**2.60**	0.97	0.0	18.90
Mulernes2	27.60	12.51	0.0	**1.30**	1.16	0.0	26.30
NZahles1	97.40	28.46	2.2	**64.50**	0.71	2.0	32.90
NZahles2	111.20	35.32	0.9	**72.60**	6.10	0.0	38.60
NielsSteensen1	309.80	42.98	3.0	**214.40**	38.20	3.0	95.40
NielsSteensen2	124.10	24.92	0.0	**90.50**	22.65	0.0	33.60

Table 5 continued

	Lectio SA			ALNS			
	$\bar{x}$	σ	#UA	$\bar{x}$	σ	#UA	Diff.
NielsSteensen3	186.40	37.65	5.0	**147.60**	22.31	5.0	38.80
Nordsjaelland1	1036.30	74.52	4.0	**652.20**	55.28	4.0	384.10
Nordsjaelland2	69.50	2.17	586.0	**68.50**	1.58	586.0	1.00
Nordsjaelland3	**1681.40**	132.07	45.0	1927.20	191.16	45.0	-245.80
Nordsjaelland4	**1627.80**	135.70	65.7	2335.90	270.87	65.8	-708.10
Nordsjaelland5	1699.10	118.08	27.6	**1104.40**	76.82	27.0	594.70
Nordsjaelland6	1369.00	102.13	32.1	**1109.90**	61.44	32.0	259.10
Nordsjaelland7	1646.40	106.74	49.1	**1567.70**	87.35	49.0	78.70
Nordsjaelland8	2065.30	179.20	37.0	**1736.00**	93.59	37.0	329.30
Nyborg1	89.70	16.45	0.0	**72.10**	0.32	0.0	17.60
Nyborg2	93.50	23.75	0.0	**51.00**	1.49	0.0	42.50
Nyborg3	0.40	0.70	55.6	**0.00**	0.00	55.0	0.40
Nyborg4	14.40	6.75	3.0	**1.50**	0.85	3.0	12.90
Naerum	424.50	43.07	9.0	**248.40**	50.63	9.0	176.10
Noerresundby1	36.10	18.11	0.0	**9.90**	3.54	0.0	26.20
Noerresundby2	115.70	26.60	0.3	**34.90**	10.12	0.0	80.80
Odder	69.80	27.68	6.0	**21.00**	0.00	6.0	48.80
~~Oure~~	~~40.80~~	~~11.13~~	~~15.7~~	~~24.40~~	~~8.22~~	~~14.0~~	~~N/A~~
Paderup	141.50	15.81	0.0	**75.50**	4.48	0.0	66.00
Randers1	168.40	22.17	0.0	**93.50**	0.53	0.0	74.90
Randers2	244.00	77.90	0.0	**10.80**	3.79	0.0	233.20
Rosborg1	0.00	0.00	246.0	0.00	0.00	246.0	0.00
~~Rosborg2~~	~~2611.00~~	~~299.16~~	~~19.1~~	~~837.10~~	~~75.08~~	~~18.0~~	~~N/A~~
Roskilde	202.30	60.81	0.0	**139.00**	23.25	0.0	63.30
Rybners1	274.90	45.42	0.0	**174.90**	1.10	0.0	100.00
Rybners2	1043.80	67.33	15.0	**789.10**	4.15	15.0	254.70
Roedkilde1	10.90	4.84	0.0	**2.00**	1.33	0.0	8.90
Roedkilde2	19.30	7.48	7.0	**12.00**	2.67	7.0	7.30
Roedkilde3	57.10	11.75	0.0	**6.80**	2.35	0.0	50.30
~~SanktAnnae1~~	~~205.90~~	~~57.43~~	~~60.3~~	~~201.30~~	~~22.07~~	~~51.6~~	~~N/A~~
SanktAnnae2	0.00	0.00	7.0	0.00	0.00	7.0	0.00
Skanderborg1	3.50	1.65	0.0	**0.00**	0.00	0.0	3.50
Skanderborg2	35.40	34.59	0.9	**5.80**	3.79	1.0	29.60
Skive	176.50	47.52	18.0	**89.00**	25.05	18.0	87.50
Slagelse1	130.80	5.94	3.0	**114.90**	5.49	3.0	15.90
Slagelse2	29.70	19.44	0.2	**3.50**	2.55	0.0	26.20
Soroe	2117.10	111.78	14.5	**883.20**	49.02	13.5	1233.90
Stoevring1	17.50	4.93	0.0	**2.70**	1.42	0.0	14.80
Stoevring2	229.20	55.04	0.0	**16.40**	4.14	0.0	212.80
Stoevring3	260.60	71.41	0.0	**14.30**	2.36	0.0	246.30
Stoevring4	124.70	48.48	0.0	**13.80**	2.25	0.0	110.90
Stoevring5	65.90	22.83	6.0	**10.30**	2.54	6.0	55.60
Varde	867.40	152.80	10.0	**326.20**	9.46	10.0	541.20
Vejen	0.30	0.48	0.0	**0.00**	0.00	0.0	0.30
ViborgG	267.90	8.84	94.0	**257.80**	1.03	94.0	10.10
ViborgH	102.20	28.61	0.0	**22.60**	9.16	0.0	79.60
~~Viby~~	~~236.30~~	~~3.65~~	~~186.8~~	~~230.60~~	~~0.52~~	~~181.8~~	~~N/A~~
Average	431.53	47.78	33.9	319.50	20.14	32.5	133.60

Each algorithm is ran 10 times on each dataset. For each algorithm and each dataset is listed the mean objective ”$\bar{x}$”, standard deviation of objective “σ”, and the number of unassigned meeting requests “#UA”. Notice that the objectives are given in the domain of the SA algorithm, which is of different magnitude than the objective of this article. Those datasets where the number of unassigned requests differs between the algorithms with more than ±1 are struck out, as this is not considered a fair comparison. Column “Diff” is the difference between mean objectives must lie in the interval ±1. This means the comparison of solvers in Table 5 is only approximate, however it can be considered as a very good approximation, since a difference of one fulfilled meeting request will have minor influence of the objective. Notice that the objectives are given in the domain of the SA algorithm, which is of different magnitude than the objective of this article. This is due to the fact that the undocumented heuristic aims at minimizing whereas the approach of this paper is to maximize.

Given the average objective of the SA algorithm $\bar{x}_{\text{SA}}$ and the average objective of the ALNS algorithm $\bar{x}_{\text{ALNS}}$, and that the comparison of these two is valid, we compute the difference $\bar{x}_{\text{SA}} - \bar{x}_{\text{ALNS}}$. For almost every instance where comparison is valid, the ALNS algorithm in average finds a better solution. Furthermore the solutions from the ALNS algorithm has far lower deviation than the SA algorithm, which are important, as the users of the algorithm will usually only run the algorithm once.

By the computational tests of this section, it has been shown that the ALNS algorithm is the best solution procedure, of those considered in this paper, for the CTP. It outperforms both Gurobi and the existing heuristic of Lectio in terms of both solution quality and reliability.

7 Final remarks and outlook

It has been shown how the CTP, an important real-life problem for the Danish high schools, can be modeled using linear IP. ALNS has proven successful in establishing solutions for two versions of the problem, the PCTP and the SCTP. Furthermore, F-Race has shown to be an efficient method for tuning of the free parameters. The developed ALNS algorithm has been implemented in Lectio and is hence available for 95 % of the Danish high schools.

In case of the PCTP, it has been shown that the ALNS algorithm in average finds solutions which are less than 4 % from optimum. This average is taken over 100 real-life dataset, and therefore we have high confidence in this result. Furthermore it has been shown that comparing with the existing algorithm in Lectio, which is the only other known heuristic algorithm for the problem, the ALNS algorithm is far superior. For 83 of the 86 datasets, ALNS finds better solutions, and in many cases the solution quality of the ALNS is considerably better. For the remaining 14 datasets a comparison was not considered fair.

The performance for the SCTP is also tested on 100 real-life dataset. For these datasets, it is shown that the ALNS algorithm in average finds solutions less than 5 % from optimum.

For both the PCTP and SCTP the average solution found by ALNS is better than the solutions found by the state-of-the-art MIP solver Gurobi 5.01.

The main subject for further research is considered to be the use of Dantzig-Wolfe decomposition and solution using Branch-and-Price. In this context, a column in the master problem could represent a meeting-plan for a student or a teacher. This would move many constraints to the subproblem, possibly giving a stronger IP formulation, which could lead to a more efficient IP-based solution approach.

Another possibility for future research is to combine the two solution methods described, i.e. using the MIP solver as a repair heuristic within the ALNS. Similar approaches are seen in Muller et al. (2011) and Prescott-Gagnon et al. (2009), with competitive results.

References

Adenso-Diaz, B., Laguna, M.: Fine-tuning of algorithms using fractional experimental designs and local search. Oper. Res. **54**(1), 99–114 (2006)

Azi, N., Gendreau, M., Potvin, J.Y.: An adaptive large neighborhood search for a vehicle routing problem with multiple trips. Technical report, CIRRELT (2010)

Balaprakash, P., Birattari, M., Stützle, T.: Improvement strategies for the f-race algorithm: sampling design and iterative refinement. In: Proceedings of the 4th International Conference on Hybrid Metaheuristics, HM'07, pp. 108–122. Springer, Berlin (2007)

Becker, S., Gottlieb, J., Stützle, T.: Applications of racing algorithms: an industrial perspective. In: Talbi, E.G., Liardet, P., Collet, P., Lutton, E., Schoenauer, M. (eds.) Artificial Evolution. Lecture Notes in Computer Science, vol. 3871, pp. 271–283. Springer, Berlin (2006)

Birattari, M.: The Problem of Tuning Metaheuristics as Seen from a Machine Learning Perspective, vol. 292: Dissertations in Artificial Intelligence—Infix, 1st edn. Springer, Berlin (2005)

Birbas, T., Daskalaki, S., Housos, E.: School timetabling for quality student and teacher schedules. J. Sched. **12**, 177–197 (2009)

Burke, E., Petrovic, S.: Recent research directions in automated timetabling. Eur. J. Oper. Res. **140**(2), 266–280 (2002). doi:10.1016/S0377-2217(02)00069-3

Carter, M., Laporte, G.: Recent developments in practical course timetabling. In: Burke, E., Carter, M. (eds.) Practice and Theory of Automated Timetabling II. Lecture Notes in Computer Science, vol. 1408, pp. 3–19. Springer, Berlin (1998)

Chiarandini, M., Birattari, M., Socha, K., Rossi-Doria, O.: An effective hybrid algorithm for university course timetabling. J. Sched. **9**, 403–432 (2006)

de Haan, P., Landman, R., Post, G., Ruizenaar, H.: A case study for timetabling in a dutch secondary school. In: Burke, E., Rudova, H. (eds.) Practice and Theory of Automated Timetabling VI. Lecture Notes in Computer Science, vol. 3867, pp. 267–279. Springer, Berlin (2007)

Diao, Y., Eskesen, F., Froehlich, S., Hellerstein, J., Spainhower, L., Surendra, M.: Generic online optimization of multiple configuration parameters with application to a database server. In: Brunner, M., Keller, A. (eds.) Self-Managing Distributed Systems. Lecture Notes in Computer Science, vol. 2867, pp. 79–93. Springer, Berlin (2003)

Erben, W., Keppler, J.: A genetic algorithm solving a weekly course-timetabling problem. In: Burke, E., Ross, P. (eds.) Practice and Theory of Automated Timetabling. Lecture Notes in Computer Science, vol. 1153, pp. 198–211. Springer, Berlin (1996)

Hutter, F., Hoos, H., Leyton-Brown, K., Stützle, T.: Paramils: an automatic algorithm configuration framework. J. Artif. Int. Res. **36**, 267–306 (2009)

Kristiansen, S., Stidsen, T.R.: Adaptive large neighborhood search for student sectioning at danish high schools. In: Proceedings of the Ninth International Conference on the Practice and Theory of Automated Timetabling (PATAT, 2012) (2012)

Kristiansen, S., Sørensen, M., Stidsen, T.R.: Elective course planning. Eur. J. Oper. Res. **215**(3), 713–720 (2011). doi:10.1016/j.ejor.2011.06.039

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

Lei, H., Laporte, G., Guo, B.: The capacitated vehicle routing problem with stochastic demands and time windows. Comput. Oper. Res. **38**(12), 1775–1783 (2011). doi:10.1016/j.cor.2011.02.007

McCollum, B.: University timetabling: bridging the gap between research and practice. In: Proceedings of the 5th International Conference on the Practice and Theory of Automated Timetabling, pp. 15–35. Springer, Berlin (2006)

Mittelman, H.: Benchmarks for optimization software. http://plato.asu.edu/bench.html (2012)

Montero, E., Riff, M.C., Neveu, B.: An evaluation of off-line calibration techniques for evolutionary algorithms. In: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, GECCO '10, pp. 299–300. ACM, New York (2010). doi:10.1145/1830483.1830540

Muller, L.: An adaptive large neighborhood search algorithm for the resource-constrained project scheduling problem. In: MIC 2009: The VIII Metaheuristics International Conference (2009)

Müller, T., Murray, K.: Comprehensive approach to student sectioning. Ann. Oper. Res. **181**, 249–269 (2010)

Muller, L.F., Spoorendonk, S.: A hybrid adaptive large neighborhood search algorithm applied to a lot-sizing problem. Tech. Report, DTU Management Engineering (2010)

Muller, L., Spoorendonk, S., Pisinger, D.: A hybrid adaptive large neighborhood search heuristic for lot-sizing with setup times. Eur. J. Oper. Res. **218**(3), 614–623 (2011)

Pellegrini, P., Birattari, M.: Implementation effort and performance. In: SLS 2007, pp. 31–45. Springer, Berlin (2007)

Pellegrini, P., Stützle, T., Birattari , M.: Off-line vs On-line Tuning: A Study on Max–Min Ant System for the TSP. In: Swarm Intelligence. Lecture Notes in Computer Science, vol. 6234, pp. 239–250. Springer, Berlin (2010)

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

Pisinger, D., Ropke, S.: A general heuristic for vehicle routing problems. Comput. Oper. Res. **34**, 2403–2435 (2005)

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

Potvin, J.Y., Rousseau, J.M.: A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. Eur. J. Oper. Res. **66**(3), 331–340 (1993)

Prescott-Gagnon, E., Desaulniers, G., Rousseau, L.M.: A branch-and-price-based large neighborhood search algorithm for the vehicle routing problem with time windows. Networks **54**(4), 190–204 (2009). doi:10.1002/net.20332

Ribeiro, G.M., Laporte, G.: An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem. Comput. Oper. Res. **39**(3), 728–735 (2012). doi:10.1016/j.cor.2011.05.005

Ropke, S., Pisinger, D.: An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transp. Sci. **40**, 455–472 (2006)

Santos, H., Uchoa, E., Ochi, L., Maculan, N.: Strong bounds with cut and column generation for class-teacher timetabling. Ann. Oper. Res. **194**, 399–412 (2010). doi:10.1007/s10479-010-0709-y

Schaerf, A.: A survey of automated timetabling. Artif. Intell. Rev. **13**, 87–127 (1999). doi:10.1023/A:1006576209967

Shaw, P.: A new local search algorithm providing high quality solutions to vehicle routing problems. Working Paper, University of Strathclyde, Glasgow (1997)

Shaw, P.: Using constraint programming and local search methods to solve vehicle routing problems. In: Maher, M., Puget, J.F. (eds.) Principles and Practice of Constraint Programming-CP98. Lecture Notes in Computer Science, vol. 1520, pp. 417–431. Springer, Berlin (1998)

Sørensen, M., Stidsen, T.R.: 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) (2012)

Tripathy, A.: School timetabling—a case in large binary integer linear programming. Manag. Sci. **30**(12), 1473–1489 (1984)