Computers & Operations Research 151 (2023) 106089

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor

An adaptive large neighbourhood search metaheuristic for hourly learning activity planning in personalised learning

Niels A. Wouda *, Ayse Aslan, Iris F.A. Vis

Department of Operations, University of Groningen, Nettelbosje 2, 9747 AE, Groningen, The Netherlands

ARTICLE INFO

Keywords:
Personalised learning
OR in education
Timetabling
Adaptive large neighbourhood search
Metaheuristic
Secondary education

ABSTRACT

Personalised learning offers an alternative method to one-size-fits-all education in schools, and has seen increasing adoption over the past several years. Personalised learning's focus on learner-driven education requires novel scheduling methods. In this paper we introduce the hourly, learner-driven activity planning problem of personalised learning, and formulate scheduling methods to solve it. We present an integer linear programming model of the problem, but this model does not generate schedules sufficiently quickly for use in practice. To overcome this, we propose an adaptive large neighbourhood search metaheuristic to solve the problem instead. The metaheuristic's performance is compared against optimal solutions in a large numerical study of 14,400 instances. These instances are representative of secondary education in the Netherlands, and were developed from expert opinions. Solutions on average deviate only 1.6% from optimal results. Further, our experiments numerically demonstrate the mitigating effects changes to the structure and staffing of secondary education have on the challenges of satisfying learner instruction demands in personalised learning.

1. Introduction

Traditional methods to organise secondary education fix schedules and learning goals on an annual or semi-annual basis. Such schedules do not emphasise individual competencies, but rather the predictable performance of the entire learner group. In education, this so-called 'class-teacher model' (Asratian and De Werra, 2002) poses problems for learners in deviation of the average.

Personalised learning (PL) offers a shift away from this traditional model, towards a learner-oriented perspective on education. Although a singular definition of PL is lacking, emphasis is placed on learner-driven instructional objectives, content, pace, and sequencing. Activities are frequently learner-initiated, with learners receiving instruction in the material based on learner demands, allowing learning paths and curriculum pacing to be adjusted to each learner's needs. This description matches that of the Swedish *Kunskapsskolan*, which expressly aims to facilitate PL (Eiken, 2011). The *Kunskapsskolan* model, introduced in 2000, is now being implemented in more than 100 schools around the world (Kunskapsskolan, 2019). In Europe, the European Commission regards PL as an important strategy to transform education (European Political Strategy Centre, 2019). The individual nature of PL poses challenges for school timetabling: dynamic learner demands prove a poor fit for the long-term schedules that are generated by most classical scheduling methods. PL calls for new, short-term scheduling methods based on learner demands.

This paper introduces the hourly learning activity planning problem (HLAPP) of PL schools, and a metaheuristic procedure for solving the problem. In the HLAPP, learning activities on specific learning topics are formed every hour by assigning suitable teachers and classrooms based on individual learner demands. In PL schools, when learner demands for activities are known for a longer period (e.g., for a week), a static longer-term initial learning activity plan can be used. The HLAPP aims to supplement these longer-term initial plans at times in which there is a need for dynamic (re)scheduling based on changes in learner demands or in the availability of teachers and classrooms. Based on these changes, an initial plan must be updated promptly to reflect the new state of learner demands (see Veenstra and Vis, 2016 for scheduling under perturbances in classical timetabling). Hourly and weekly learning activity planning does not only differ in the planning horizon. When the length of the planning horizon becomes longer than an hour (e.g., daily or weekly), plans should take into account constraints relating to the sequence of activities assigned to each learner. For example, in Aslan et al. (2020), where a weekly activity plan is formed, course activities must be assigned to learners sequentially. In contrast, HLAPP instead focuses on the aspect of assigning learners to their most preferred activities in a given hour, where the sequencing constraints are presented implicitly by the course activities learners opt to take. Due to the hourly aspect of HLAPP, it is particularly important

* Corresponding author.
E-mail addresses: n.a.wouda@rug.nl (N.A. Wouda), ayse.aslan@rug.nl (A. Aslan), i.f.a.vis@rug.nl (I.F.A. Vis).

https://doi.org/10.1016/j.cor.2022.106089
Received 20 September 2021; Received in revised form 14 September 2022; Accepted 15 November 2022
Available online 19 November 2022

to obtain solutions quickly, within just a few minutes. We propose an efficient adaptive large neighbourhood search (ALNS) metaheuristic for solving the HLAPP. ALNS approaches are successful in various scheduling problems (Wen et al., 2016; Lei et al., 2018). Our metaheuristic complements longer-term timetabling methods that provide an initial schedule; it resolves in real-time any updates to learner demands and other inconsistencies that occur as the initial schedule unfolds.

School timetabling problems typically studied in the literature concentrate on traditional educational models. To the best of our knowledge, there are only a few studies which explore learner-centred timetabling problems. Santiago-Mozos et al. (2005) present a student-preference based course timetabling problem in a Spanish university. Kristiansen et al. (2011) study a student-centred elective course planning problem in Danish high schools. Kannan et al. (2012) propose a multi-stage graph-theoretic approach to the scheduling problem of a group of personalised learning schools in New York City. In all of these studies, learners provide preferences or demands over a set of courses at the beginning of a semester or a year, to be considered in the longer-duration timetables. Both Pillay (2014) and Johnes (2015) note the increased use of metaheuristics in high school timetabling in recent years. Neither, however, mention ALNS as a metaheuristic for the timetabling problem, despite several papers finding good results (Sørensen et al., 2012; Sørensen and Stidsen, 2012; Kristiansen et al., 2013). More recently, Kiefer et al. (2017) also observe good performance with an ALNS-based approach for a static course timetabling problem.

The rest of this paper is organised as follows. Section 2 formally describes the HLAPP. In Section 3 we provide an integer linear program for the HLAPP, and in Section 4 we develop our metaheuristic solution approach. We compare the performance of this approach to optimal solutions through a series of numerical experiments based on Dutch secondary education in Section 5. Finally, Section 6 concludes the paper and presents suggestions for future research.

2. Problem definition

In a broad classification, PL schools form a collection of four basic resources: a set of learners $l \in L$ who demand learning activities, teachers $t \in T$ to instruct the activities, course modules $m \in M$ that define the precise learning topics, and finally classrooms $c \in C$ where the activities are scheduled. The HLAPP produces hourly learning activity plans based on learner demands by assigning these resources to form the demanded learning activities. For this, the HLAPP uses the most up-to-date learner demands for modules $\left(D_{lm} \in \mathbb{R}_{\geq 0}\right)$, with the objective of assigning each learner l to an appropriate module m such that the learners are assigned to their most demanded learning topics. Such an assignment is modelled with the decision variable $y_{lm} \in \{0, 1\}$. As a convenience, $D_{lm} = 0$ expresses that a learner l is ineligible for module m.

If a suitable teacher t and classroom c are available for a group of learners, we call the assignment of the group of learners to this classroom–teacher pair an *activity*. We distinguish two types of activities: instruction activities where a group of learners with demand for the same module receive instruction from a qualified teacher, and self-study activities where learners work independently on different modules, with a teacher present for supervision only. We assume an instruction activity by a qualified teacher results in better learner demand satisfaction than self-study activities. In particular, we say that a self-study activity satisfies only a fraction w ($0 \leq w \leq 1$) of the demand D_{lm} a learner l has for a module m, where w is a self-study penalty parameter that schools can use to balance self-study and instruction assignments in the hourly plan. Our models can also easily be extended to the case where w varies by learner and module.

It is helpful for our modelling in later sections to track self-study assignment through a dedicated module. We let $m_S \in M$ denote this self-study activity module. The demand for m_S is given by $D_{lm_S} = w \max_{m \in M \setminus m_S} D_{lm}$ for each learner l, since in a self-study assignment learners are assumed to work on their most-preferred module.

The decision variables $x_{mct} \in \{0, 1\}$ track activities of modules $m \in M$, classrooms $c \in C$, and teachers $t \in T$. Not all teachers and classrooms can be paired with each module m. Whether teacher t is qualified for module m is governed by the binary qualification matrix $Q^T_{tm} \in \{0, 1\}$. All teachers are qualified to supervise a self-study activity, that is, $Q^T_{tm_S} = 1$ for each teacher t. Similarly, not all classrooms c are equipped to host an activity for module m, which is tracked by the binary qualification matrix $Q^C_{cm} \in \{0, 1\}$.

Due to the seating capacities $N_c \in \mathbb{N}$ of classrooms $c \in C$, there is a natural upper bound on the maximum number of learners that can attend the same activity. Additionally, the HLAPP considers two additional bounds on the activity size: $\delta^- \in \mathbb{N}$ specifies the minimum number of learners needed for an activity to be scheduled, and $\delta^+ \in \mathbb{N}$ the maximum number of learners that can attend an instruction activity. The lower bound δ^- prevents very small activities from being scheduled. Such activities might not be considered an efficient use of classroom space or teacher hours. The upper bound allows one to regulate maximum instruction activity sizes, in addition to the classroom capacity. Such maximum instruction activity sizes commonly differ between education programs. For self-study activities, δ^+ is not enforced.

Table 1 summarises the notation introduced above.

3. Model

The formulation for the HLAPP is as follows. We aim to optimise the objective

$$\max_{x,y} \quad \sum_{l \in L} \sum_{m \in M} D_{lm} y_{lm} \tag{1}$$

subject to

$$\sum_{l \in L} y_{lm} \leq \sum_{c \in C} \min\left(\delta^+, N_c\right) \sum_{t \in T} x_{mct} \quad \forall m \in M \setminus \{m_S\} \tag{2}$$

$$\sum_{l \in L} y_{lm_S} \leq \sum_{c \in C} N_c \sum_{t \in T} x_{m_S ct} \tag{3}$$

$$\sum_{l \in L} y_{lm} \geq \delta^- \sum_{c \in C} \sum_{t \in T} x_{mct} \quad \forall m \in M \tag{4}$$

$$\sum_{m \in M} y_{lm} = 1 \quad \forall l \in L \tag{5}$$

$$\sum_{m \in M} \sum_{c \in C} x_{mct} \leq 1 \quad \forall t \in T \tag{6}$$

$$\sum_{m \in M} \sum_{t \in T} x_{mct} \leq 1 \quad \forall c \in C \tag{7}$$

$$\sum_{c \in C} x_{mct} \leq Q^T_{tm} \quad \forall t \in T,\ \forall m \in M \tag{8}$$

$$\sum_{t \in T} x_{mct} \leq Q^C_{cm} \quad \forall c \in C,\ \forall m \in M \tag{9}$$

$$y_{lm} \leq \mathbf{1}_{D_{lm} > 0} \quad \forall l \in L,\ \forall m \in M \tag{10}$$

$$x_{mct} \in \{0, 1\} \quad \forall m \in M,\ \forall c \in C,\ \forall t \in T \tag{11}$$

$$y_{lm} \in \{0, 1\} \quad \forall l \in L,\ \forall m \in M \tag{12}$$

The objective (1) is to assign learners to their most demanded modules. Constraint (2) ensures the number of learners assigned to a module for an instruction activity does not exceed the capacity of the assigned classrooms. Since this involves instruction activities, the capacity of each classroom c is given as the minimum of δ^+ and N_c. Constraint (3) ensures the same for self-study activities, where δ^+ does not play a role. Constraint (4) guarantees the minimum group size is met for all activities, whether they be instruction or self-study. Constraint (5) assigns each learner to a module. Constraints (6) and (7) ensure teachers and classrooms are assigned to at most one activity, respectively. Constraints (8) and (9) guarantee only qualified teachers and classrooms

Table 1
Definitions of inputs and decision variables.

Notation	Definition
$l \in L$	Set of learners.
$c \in C$	Set of classrooms.
$t \in T$	Set of teachers.
$m \in M$	Set of modules, including m_S, the self-study module.
$D \in \mathbb{R}_{\geq 0}^{\lvert L \rvert \times \lvert M \rvert}$	Demand matrix. D_{lm} specifies the demand of learner $l \in L$ for module $m \in M$. This demand is 0 when the learner is ineligible for the module.
$Q^T \in \{0,1\}^{\lvert T \rvert \times \lvert M \rvert}$	Teacher qualification matrix. $Q^T_{tm} = 1$ if teacher $t \in T$ can teach module $m \in M$, 0 otherwise.
$Q^C \in \{0,1\}^{\lvert C \rvert \times \lvert M \rvert}$	Classroom qualification matrix. $Q^C_{cm} = 1$ if classroom $c \in C$ can host an activity for module $m \in M$, 0 otherwise.
$w \in [0,1]$	Self-study penalty to balance instruction and self-study assignments.
$\delta^- \in \mathbb{Z}_{\geq 0}$	Minimum activity size.
$\delta^+ \in \mathbb{Z}_{\geq 0}$	Maximum instruction activity size.
$N_c \in \mathbb{N}$	Capacity of classroom $c \in C$.
$x_{mct} \in \{0,1\}$	Decision that is 1 if an activity with module $m \in M$ is scheduled in classroom $c \in C$, taught by teacher $t \in T$, 0 otherwise.
$y_{lm} \in \{0,1\}$	Decision that is 1 if learner $l \in L$ is assigned to module $m \in M$, 0 otherwise.

are assigned for each module activity. Constraint (10) ensures learners l are assigned only to modules m they are eligible to take, using an indicator that is 1 when $D_{lm} > 0$ and 0 otherwise. Finally, (11) and (12) are boxing constraints on the binary variables x and y.

The decision variables x and y produce an implicit schedule, as they assign learners to modules (through y) and ensure sufficient teachers and classrooms are available for activities to be planned for the given number of learners and modules (through x). We turn this into an explicit schedule that can be used by schools by 'filling' the activities x with learners in a simple post-processing step. First, the minimum group size is met for all classroom–teacher pairs assigned to a module by assigning δ^- learners of the learners demanding this module to each classroom–teacher pair. Then, all remaining module learners are scheduled into these pairs in a second pass, respecting the upper bounds δ^+ and N_c for each activity.

The formulation of (1)–(12) combines a simple assignment problem involving the variables y with feasibility side constraints involving the three-dimensional variables x. These variables x complicate the formulation because they turn the problem into a variant of the three-dimensional assignment problem, which is well-known to be NP-hard. As such, we explore an alternative, metaheuristic solution strategy in Section 4.

4. Solution approach

We propose an adaptive large neighbourhood search (ALNS) metaheuristic for our problem. Røpke and Pisinger (2006) originally introduced the ALNS procedure for vehicle routing problems, and a recent, general treatment is given in Pisinger and Røpke (2019). Pseudo-code for the metaheuristic is given in Algorithm 1. In this section we describe how to design the building blocks of our metaheuristic for solving the HLAPP.

In general, the ALNS metaheuristic consists of several steps. The algorithm begins with an initial solution. This solution should be feasible, but need not be very good. Then the algorithm iterates for a fixed number of iterations. In each iteration, it selects a destroy and repair operator from the operator collection (O_D for destroy and O_R for repair operators, respectively), which transform the current solution s into a candidate solution s^c. This candidate solution is then evaluated, and the operator selection mechanism is updated based on the evaluation outcome.

The rest of this section is structured as follows. We first propose an initial, constructive solution for the HLAPP. Then we describe a set of destroy and repair operators tailored to our problem. We then explain the acceptance criterion we use to determine if the candidate solution should replace the current and best solutions. If the candidate solution is a new best solution, we apply a local search (LS) procedure to further improve the new solution. Finally, we describe the *adaptive* part of our ALNS metaheuristic, which uses updating weights to select the destroy and repair operators.

Algorithm 1: Adaptive large neighbourhood search.

Input : Initial feasible solution s.
Output: Best observed solution s^*.
$s^* := s$, $\rho_D := (1, \dots, 1)$, $\rho_R := (1, \dots, 1)$.
repeat
  Select destroy and repair methods $d_{\text{op}} \in O_D$, $r_{\text{op}} \in O_R$ using ρ_D and ρ_R.
  $s^c := r_{\text{op}}(d_{\text{op}}(s))$
  if s^c is accepted **then**
    $s := s^c$
  if s^c has a better objective value than s^* **then**
    $s^* :=$ Local-Search(s^c)
    $s := s^*$
  Update ρ_D and ρ_R
until maximum number of iterations is exceeded
return s^*

Initial solution. The initial solution assigns all learners to self-study activities. These activities are formed by selecting a random teacher t and qualified classroom c, and inserting up to N_c learners into the resulting activity. Since all teachers are qualified to supervise self-study activities, there is no need to test their qualification. Although it is theoretically possible that there are insufficient qualified classrooms to assign all learners to self-study activities, that concern seems largely theoretical: we have never observed a situation where no feasible initial solution existed, and it seems unlikely that such school buildings actually exist in practice.

Destroy operators. The destroy operators each remove approximately $d > 0$ learners from their current assignments.

1. *Random activity removal*
   This operator randomly removes entire *activities* (recall that an activity is an assignment of groups of learners to a single classroom, teacher and module) from the solution. All learners in a removed activity are marked unassigned, and the activity's teacher and classroom are also unassigned. The procedure repeats until at least d learners have been removed.
2. *Smallest activity removal*
   This operator removes small activities (in terms of number of learners assigned) until at least d learners have been removed

from the solution. The motivation for this operator is that it unassigns classrooms and teachers that are used by only a small number of learners, potentially freeing them for use in activities with more learners.

3. *Random learner removal*
   This operator randomly removes a learner from an activity and marks it as unassigned. This is only done if the learner can feasibly be removed, that is, such that at least δ^- learners remain in the activity after this learner is removed. The procedure repeats until d learners have been removed from their respective activities, or until no more learners can be removed without rendering the remaining activities infeasible.
4. *Worst regret learner removal*
   This operator removes a set of learners from the solution that are least met in their demands. The rationale for this operator is that it removes learners whose scheduling decisions incur significant regret. We compute this regret as the difference between the best and current assignments for each learner. Formally, for learner $l \in L$ assigned to module $m \in M$ in the current solution, we compute the regret as

   $$r_{lm} = \max_{m' \in M} \left\{ D_{lm'} \right\} - D_{lm},$$

   where larger regrets are worse.
   We sort these regrets in decreasing order, and keep track of the associated learners. A total of d learners are selected using a skewed distribution, which favours larger over smaller regrets. Our distribution is (decreasing) triangular for the first d values, and uniformly flat thereafter, for all costs with index in $\{d + 1, \dots, |L|\}$. A learner l with cost at index $j \in \{1, \dots, |L|\}$ then has probability

   $$\Pr(\text{select } l) = \begin{cases} \frac{d-j+1}{\sum_{i=1}^{d} i+|L|-j} & \text{if } j \leq d, \\ \frac{1}{\sum_{i=1}^{d} i+|L|-j} & \text{otherwise,} \end{cases}$$

   of being selected. Each of the selected d learners is removed from the solution, if this leaves behind at least δ^- learners in their respective activities.

Repair operators. Each repair operator re-inserts all unassigned learners into the solution, such that the resulting solution is feasible.

1. *Break-out activity*
   This operator creates new activities from the unassigned learners. All such learners are grouped by the modules they demand. The modules are ordered by decreasing aggregated learner demand. For each such module, if an activity can be scheduled (δ^- is respected, and a qualified classroom and teacher are available), all grouped learners are reassigned to the new activity. The operator attempts to select second-degree teachers for second-degree modules, rather than use a first-degree teacher for a second-degree module. Similarly, for classrooms it attempts to select the smallest classroom where the grouped learners fit: if this turns out to be impossible, the largest available classroom is selected instead, and not all learners will be scheduled into this activity.
   When a new activity is scheduled, the operator also attempts to schedule self-study learners into the new activity, if that is an improvement over their current self-study assignments. Finally, if no new activities can be scheduled and some learners remain unassigned, *greedy learner insert* is applied to the remaining learners.
2. *Greedy learner insert*
   This operator randomly selects an unassigned learner and inserts the learner into the best feasible instruction activity. Such an insertion is feasible only when the instruction activity has less than $\min\{\delta^+, N_c\}$ learners assigned, and when the selected learner demands the activity's module. If no feasible instruction activity exists, the learner is assigned to a self-study activity, or a new self-study activity is created. In rare cases that too might not be possible: when this happens, the instruction activity with the smallest contribution to the objective value is converted into self-study, and the learner is assigned there. The procedure repeats until all unassigned learners have been assigned to an activity.

Acceptance criterion. A new solution s^c is accepted based on a simulated annealing (SA) procedure, motivated by the good results Santini et al. (2018) find using this acceptance criterion in ALNS metaheuristics for various problems. The SA criterion works as follows. If the new solution s^c is better than the current solution s, it is always accepted. Otherwise, it is accepted with probability

$$\Pr(\text{accept } s^c) = \exp\left\{ \frac{f(s^c) - f(s)}{T} \right\},$$

with T the temperature in the current iteration, and $f(\cdot)$ the objective value function. The temperature T is set to the starting temperature at the beginning of the metaheuristic procedure, and then decreased in every iteration by multiplying it with a cooling rate parameter $\gamma \in (0, 1)$ until a final temperature is reached.

Local search (LS). When a new best solution is found, a *reinsert learner* operator is applied to improve the solution further. This operator first determines all improving relocation moves between activities that learners can make, given the activities that are currently scheduled in the solution. It then applies these moves in order of decreasing objective gain (best moves first). If a move has been rendered infeasible because another learner has moved previously, the move is skipped. The operator is then called again on the improved solution, which is repeated until no further improving moves are found.

Updating and operator selection. We apply the same roulette wheel mechanism when determining which operator to select as outlined in Røpke and Pisinger (2006). Assuming there are $k \in \mathbb{N}$ operators, each with weights $z_i \geq 0$, $i \in \{1, \dots, k\}$, the probability of selecting operator j is given by

$$\Pr(\text{select } j) = \frac{z_j}{\sum_{i=1}^{k} z_i}.$$

We maintain two lists of such weights: one for the destroy operators (ρ_D), and one for the repair operators (ρ_R). At the start of the algorithm, these weights are all initialised to 1.

To update the weights of the destroy and repair operators, we determine their performance based on three possible outcomes: (i) a new best solution is found, (ii) the current solution is improved, but the global best solution remains unchanged, and (iii) the solution is accepted as the new one, without improving its objective. Each outcome is assigned a factor ω_i, $i \in \{1, \dots, 3\}$. For a given operator j and observed outcome i, the weights are updated as a convex combination of the original weight and the observed outcome, as

$$z_j := \theta z_j + (1 - \theta)\omega_i,$$

using a decay parameter $\theta \in [0, 1]$, which controls how quickly the metaheuristic responds to changes in the effectiveness of its operators. As we cannot differentiate the effect of the destroy and repair operators in a single iteration, both are updated by the same factor.

5. Experiments

We turn to numerical experiments to validate our proposed metaheuristic. In addition, we investigate various policy decisions regarding the number and composition of classrooms and teachers. In contrast to traditional school timetabling (Post et al., 2012), no standardised benchmarks yet exist for PL in the Dutch setting. As such, we formulate our own set of benchmark cases based on expert estimates.

Table 2
Experiment parameters and their levels.

Parameter	Levels
Self-study penalty parameter (ω)	50%, 75%
Demand spread (σ)	0, 1, 2, 3
School size (# learners)	800, 1200, 1600
Teacher qualification distribution	$(1; 0; 0)$, $(0.5; 0.5; 0)$, $(0.4; 0.4; 0.2)$
Instruction classrooms and capacities	Regular number of classrooms of 32 capacity, double the number of classrooms of 16 capacity.
Minimum activity size (δ^-)	5
Maximum instruction activity size (δ^+)	30

Our experimental design is given in Section 5.1. In Section 5.2, the metaheuristic performance is compared with the model outcomes, and policy effects are discussed.

5.1. Experimental design

Each experimental instance consists of a complete description of all parameters required by the exact model of Section 3 and metaheuristic of Section 4. We define classrooms, course modules, teachers and learners to represent a six-year secondary education program in the Netherlands. With these experiments we investigate the effects of all parameter level combinations given in Table 2 (full factorial design). These combinations result in a total of 144 experiments of 100 instances each, in line with Veenstra and Vis (2016).

We consider medium (M), large (L) and extra large (XL) school sizes with the following characteristics:

- M: 800 learners, 80 teachers, 40 instruction, and 3 large self-study classrooms.
- L: 1200 learners, 120 teachers, 60 instruction, and 4 large self-study classrooms.
- XL: 1600 learners, 160 teachers, 80 instruction, and 6 large self-study classrooms.

Each instruction classroom has a capacity of 32 learners, and a self-study classroom a capacity of 80 learners. Finally, each school is assumed to teach 12 courses of 48 modules each (nominally 8 per year).

Since learners under the PL paradigm may work on a number of different modules at any time, it is likely that group sizes for instruction activities related to a specific module are considerably smaller than those in classical methods of educational organisation, where fixed year groups remain common. As such, it is expected that most contemporary school buildings feature classrooms with capacities that exceed the needs for PL instruction activities. Motivated by this observation, we investigate a policy where all instruction classrooms are split in two. This may be achieved within existing school buildings via *e.g.* the installation of partition walls. Note that the aggregated classroom capacity is unchanged between policies.

We set δ^- to 5 and δ^+ to 30, for all experiments. These parameters are based on expert insight into traditional timetabling, and not expected to change under the PL paradigm.

Teachers in Dutch secondary education are qualified per course according to a first- and second-degree structure. A first-degree teacher is allowed to teach all modules in a given course. A second-degree teacher in the same course is barred from teaching more advanced modules, in particular those of the upper three years of the six-year secondary education program. Thus, the first 24 modules of each course can be taught by teachers with either a first- or second-degree qualification, while the final 24 modules require a first-degree qualification.

We stated in Section 2 that self-study assignments do not require qualified teachers: any teacher suffices to supervise such an activity. We now introduce a third-degree teacher qualified only to supervise self-study, but not instruction activities. The qualification distribution may be expressed as a triplet of $(p; q; 1 - p - q)$ for respectively the fraction of first-degree, second-degree, and third-degree teachers in the teacher set T ($p + q \leq 1$, $p, q \geq 0$). This is the notation used in Table 2. We investigate three levels: $(1; 0; 0)$, where all teachers have a first-degree qualification, and are thus allowed to teach any and all modules within their courses; $(0.5; 0.5; 0)$, which offers a mix of second- and first-degree teachers and is commonly encountered in Dutch secondary education; and $(0.4; 0.4; 0.2)$, which introduces a small group of these third-degree teachers. Investigating the effects of such a third-degree qualification is relevant economically, as higher-qualified personnel is more expensive, and pragmatic, as there is a considerable shortage of qualified teaching personnel in Dutch secondary education in general (Ministerie van Onderwijs, Cultuur en Wetenschap, 2020). We test if the quality of schedules does not significantly alter when staff composition is varied.

Table A.3 (Appendix A) lists a representative set of courses, the classroom qualifications required for their modules, and the total number of hours learners generally spend on each course per week in current timetabling. These are given by expert estimate. We use the weekly hours to assign teacher and classroom qualifications by dividing the total number of teachers and classrooms over the courses based on the fraction of the weekly hours spent on each course. As an example: a course that is scheduled for 3 h in a 30 h schedule is assigned 10% of all teachers, and 10% of all instruction classrooms are assigned the course's room type. Given an assignment of teachers to courses, we then populate the qualification matrix Q^T based on the teacher qualification distribution. Similarly, the instruction classroom room type assignments are encoded in the classroom qualification matrix Q^C. We assume all "Regular" classrooms can be used for self-study, in addition to the dedicated, large self-study classrooms.

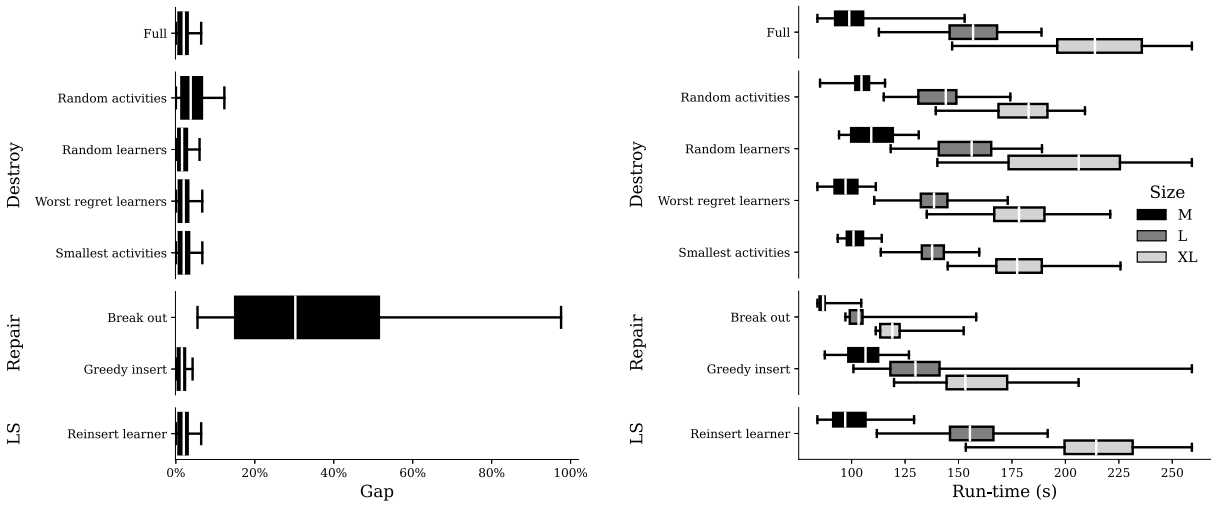
We randomly assign a learner demand to a module in each course, as follows. We take the nominal progression as a basis for module selection. This is achieved by computing the midpoint module a learner in year $y_l \in \{0, 1, \dots, 5\}$ should be working on, assuming they follow the nominal learning path of 8 modules per year. We compute $\mu_l = 8y_l + 4$. We introduce some randomness to simulate different learner aptitudes by drawing a realisation from a $\mathcal{N}(\mu_l, \sigma)$ distribution and rounding it to the nearest integer in $\{1, 2, \dots, 48\}$. The σ parameter follows from Table 2, and specifies the standard deviation of the normal distribution (for $\sigma = 0$, we have a degenerate distribution centred at μ_l). The σ parameter thus controls the measure of demand spread between learners: larger values of σ result in learner demands that are much more spread out over modules. This results, on average, in smaller instruction activities. In practice, the values of σ presented in Table 2 result in 95% of learners from the same cohort working on course modules that are within 5 (for $\sigma = 1$), 9 (for $\sigma = 2$), or 13 (for $\sigma = 3$) modules from each other. For $\sigma = 0$, all learners from the same cohort work on the same, nominal module.

For the module so obtained a demand value is drawn from an $\text{Exp}(\beta = 2)$ distribution, which is a convenient choice for a non-negative, continuous distribution. We select 50% and 75% as two levels for the self-study penalty ω.

An overview of the parameter levels for each experiment is given in Table A.4 (Appendix A).

5.2. Results

The model of Section 3 is implemented in Python 3.9 and solved with Gurobi 9.1. The ALNS metaheuristic of Section 4 is implemented



(a) Optimality gaps (%). **(b)** Run-times (seconds) by school size.

Fig. 1. Operator quality results, as box-and-whisker plots. The top row presents the performance of the complete metaheuristic, with all operators. Each following row shows the results of taking the named operator out of the metaheuristic. The whiskers indicate the bandwidth between the minimum and maximum values.

in Python 3.9 as well, using the open-source ALNS package. For the model, each experiment instance was initially given eight hours of run-time on two Intel Xeon E5 2680v3 2.5 GHz CPU cores with 32 GB of memory. Around 88% of experiment instances solved within these time and memory limits (12,710 out of 14,400 instances). The remaining 12% were given another attempt, with one day of run-time on two CPU cores, and 64 GB of memory. This solved another 11% of the instances, resulting in an overall completion rate of 99% (14,274 out of 14,400 instances). By contrast, our proposed metaheuristic has very limited resource requirements: a single processor core with 2 GB of memory and a run-time of just ten minutes is sufficient for all instances.

We present our parameter tuning in Section 5.2.1. In Section 5.2.2, we compare the ALNS metaheuristic performance with optimal results on the benchmark instances generated in Section 5.1. Finally, in Section 5.2.3, we present some practical insights, and investigate the effects on schedule quality of splitting classrooms and introducing a special third-degree teaching qualification for self-study activities.

5.2.1. Tuning

We generate a single tuning instance for each of the 144 experiments described in Section 5.1. We tune a number of parameters on these instances using SMAC3 (Lindauer et al., 2022), a Bayesian optimisation package for hyperparameter optimisation. In particular, we tune the weight factors ω_i, the decay parameter θ, and the degree of destruction d. For these parameters we consider valid configuration ranges to be in $[0, 50]$ for each ω_i, $[0.5, 1]$ for the decay parameter θ, and 10% to 50% of the learners for the degree of destruction d. The SMAC3 algorithm draws parameter configurations from these ranges, and evaluates them on a tuning instance given 1,000 iterations of the ALNS metaheuristic. We perform twenty independent runs of the SMAC3 algorithm, each lasting twelve hours. The best observed configuration amongst these twenty independent runs is given by $\omega = (21.8, 13.6, 3.8)$, $\theta = 0.8$, and $d = 15\%$ (all values rounded to one decimal), and we will use these values for our metaheuristic.

For the cooling schedule we select a starting temperature based on the initial solution, analogous to Røpke and Pisinger (2006). The initial temperature is set such that, in the first iteration, a solution with an objective up to 5% worse than the initial solution is accepted with probability 0.5 (see Roozbeh et al., 2018 for computational details). We set the number of iterations to 25,000, which balances solution quality and computation times. The cooling rate is then set such that the temperature decays to an end temperature of 1 in 25,000 iterations.

We also investigate operator quality using the 144 tuning instances. First, we solve the tuning instances to optimality using the ILP formulation of Section 3. This resulted in 142 optimally solved instances. We then solve each of these 142 tuning instances with the metaheuristic, where we take out one of the destroy, repair, or LS operators in turn. This provides insight into the relative merit of each operator. Fig. 1 visualises the results: we present optimality gaps in Fig. 1(a), and run-times in Fig. 1(b).

It is clear that the solution quality degrades significantly without the *break out* repair operator, as evidenced from Fig. 1(a). When this repair operator is not available, no new activities can be created, which has a detrimental effect on the search procedure, resulting in much worse solutions. Interestingly, the exclusion of the *greedy insert* repair operator improves the solution quality on average from 2.25% to 1.65% at the cost of 30% to 40% longer run-times, as evidenced from Fig. 1(b). This is due to the increased application of the expensive *break out* operator, which accounts for up to 40% of total run-time in the full configuration. Removing other operators generally worsens the solution quality. Of these other operators, the exclusion of the *random activity* destroy operator worsens the solution quality the most: from gaps of 2.25% on average to 4.1%.

Since the *greedy insert* repair operator does not appear to be effective and appears to worsen the solution quality, we do not use it when evaluating the heuristic's performance in the next section.

5.2.2. Metaheuristic performance

We compare the results of the metaheuristic on the instances generated in Section 5.1. The ILP model of Section 3 is used to solve the experimental instances to optimality. An overview of the data at the experiment level is given in Table B.5 (Appendix B). The heuristic solves most instances within just a few minutes, and all instances within ten. We compute percentage gaps as $(f(s^*) - f(s))/f(s)$, with $f(s^*)$ the objective value of an optimal solution s^*, and $f(s)$ the objective of the metaheuristic solution s.

The metaheuristic performs very well: across all experiment instances, it on average deviates only 1.6% from the optimal objective value. At worst, in experiment 43, the metaheuristic finds a solution with an objective 6.8% above the optimal objective value. By contrast, after ten minutes of run-time, the worst ILP gap is still 28.1% for an instance in experiment 93, although the ILP gaps are on average much better than that, at just 0.7%. That small gap is largely due to the ILP's performance on cases without any learner demand spread ($\sigma = 0$): these solve very quickly. When $\sigma > 0$, the ILP's performance behaves

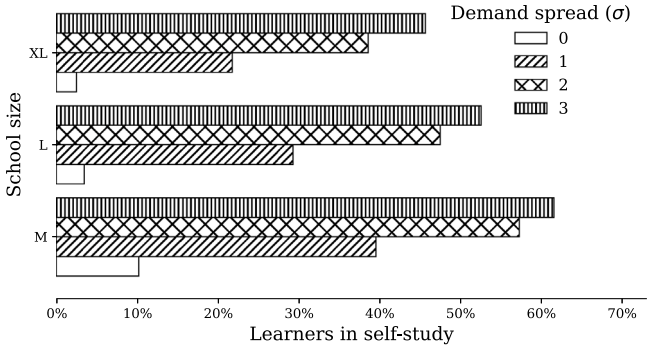


Fig. 2. Self-study learners by learner demand spread and school size.

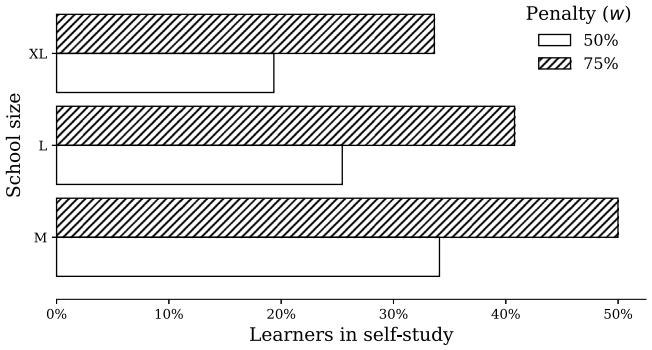


Fig. 3. Self-study learners by self-study penalty and school size.

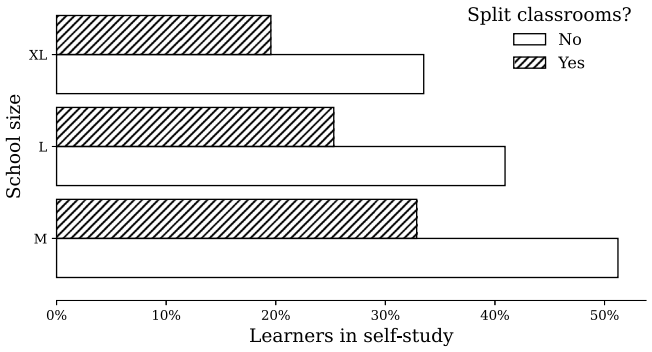


Fig. 4. Self-study learners by classroom split and school size.

more erratically, and the variation in optimality gaps after ten minutes increases (the standard deviations are 1.9% for $\sigma = 1$, 2.2% for $\sigma = 2$, and 2.6% for $\sigma = 3$). Instead, our metaheuristic offers much more predictable performance (the standard deviations are 1.2% for $\sigma = 1$, 1.3% for $\sigma = 2$, and 1.4% for $\sigma = 3$), using much less resources and runtime. Further, the metaheuristic does not rely on commercial software, which reduces the complexity of its implementation in schools.

5.2.3. Practical insights

In this section we investigate the effects of various parameters and policy choices explained in Section 5.1 on the outcomes of the experimental instances solved by the exact model of Section 3. We use the percentage of learners assigned to self-study activities as a measure of schedule quality. This is a good performance indicator, since a high percentage indicates the model had difficulty grouping learners into groups of sufficient size for an instruction activity to be scheduled, which would have avoided the self-study penalty.

Fig. 2 investigates the demand spread effect, based on the parameter σ introduced in Section 5.1. Even modest demand spread already causes

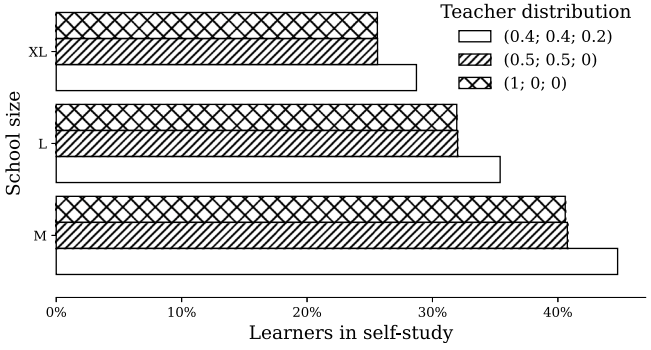


Fig. 5. Self-study learners by teacher distribution and school size.

sizeable increases in the number of learners assigned to self-study. For the medium school size this rises from on average 82 learners ($\sigma = 0$) to 316 ($\sigma = 1$), 457 ($\sigma = 2$), and reaches 491 for $\sigma = 3$. Under such a system, learners would spend anywhere between 10% to 61% of their school day in self-study activities. The effect is similar, but less pronounced in the larger schools, where even for $\sigma = 3$ around half or more of the learners are assigned to instruction activities (L: 47.6%, XL: 54.4%). This is intuitive: with the same number of modules, on average more learners have a demand for a particular module in larger schools, and can be scheduled in an instruction activity respecting the minimum activity size constraint. Our observation corroborate the findings of Aslan et al. (2020), who also report it is difficult to schedule instruction activities when learner demands are spread over many course modules, in particular in smaller schools. Given those outcomes it is interesting to study in what way changes in class room sizes and composition of the teacher group impact the amount of learners in self-study and in what way they are a potential bottleneck in implementing personalised learning.

Fig. 3 shows the effect of varying the self-study penalty w. The average number of learners in self-study activities reduces from 400 to 273 as the value of self-study is decreased from 75% to 50% for the medium school size. For larger school sizes the effect is more pronounced, with an average reduction from 489 to 305 learners for the L school, and 538 to 310 for the XL school. We conclude a self-study penalty parameter offers an effective control to balance the number of learners in self-study and instruction activities, with smaller values of w resulting in a lower number of learners in self-study activities.

Fig. 4 reports the effect of splitting classrooms. This reduces the number of learners assigned to self-study from, on average, 410 to 263 for the medium school size, on average from 491 to 304 for the large school, and 536 to 313 for the extra large school. For each school size, those numbers amount to a reduction of one third or more, which suggests splitting existing classrooms is effective at each school size.

Splitting classrooms means that large instruction activities can no longer be scheduled. We investigate whether this is a problem in practice in Fig. 6, which presents probability densities of instruction activity sizes for each demand spread level σ. It is clear that in the classical case with $\sigma = 0$, splitting classrooms does not seem beneficial: around 73% of the instruction activities are larger than the resulting classroom size of 16, and it would not be possible to schedule those activities after splitting the classrooms. For $\sigma = 1$, however, the number of instruction activities with more than sixteen learners already drops substantially to around 20% of all instruction activities. For $\sigma = 2$ only 6% of instruction activities hold more than sixteen learners, and for $\sigma = 3$ this drops to just 2.5%. Thus, in the presence of even modest demand spread, most activities can still be scheduled when splitting classrooms, and we conclude that splitting classrooms is an effective method to adapt existing school buildings for use in PL.

Fig. 5 displays the effects of our teacher qualification distribution policies. Notice that the difference between teacher distributions of

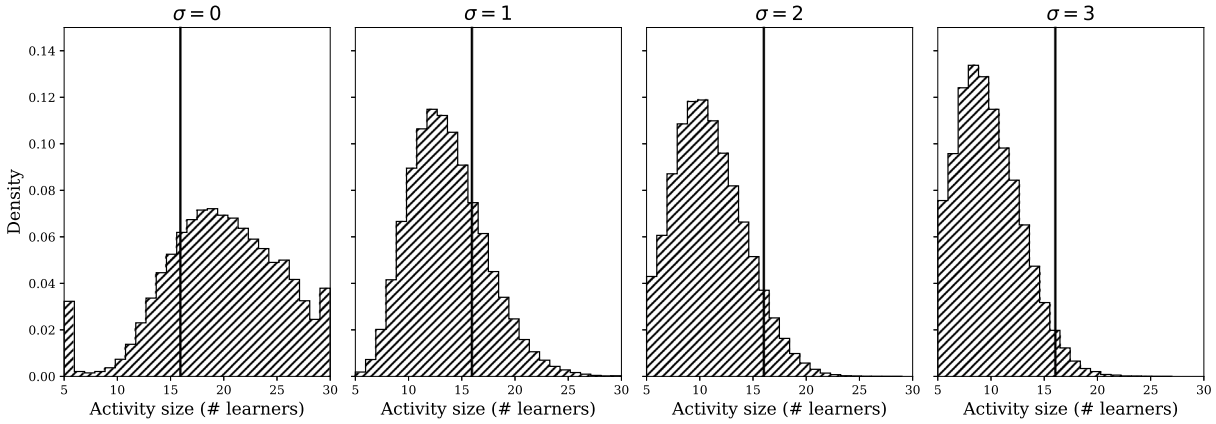


Fig. 6. Empirical densities of instruction activity sizes for instances from experiments with regular classroom sizes, for different demand spreads σ. The bumps at $\delta^- = 5$ and $\delta^+ = 30$ of the density for $\sigma = 0$ are due to the 'filling' post-processing step explained in Section 3. The solid vertical line at 16 indicates the maximum size of classrooms after splitting.

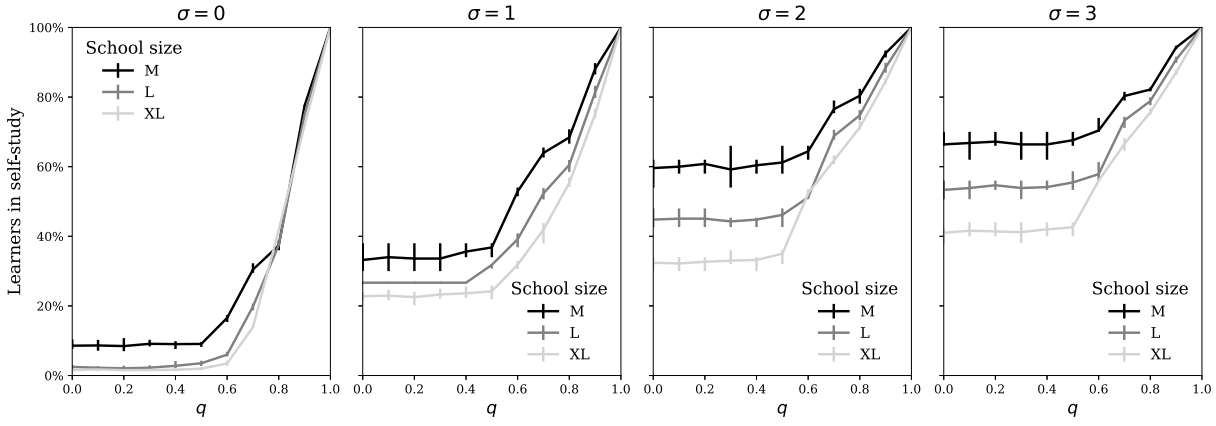


Fig. 7. Percentage of learners assigned to self-study activities as a function of q, for different demand spreads σ.

$(1; 0; 0)$ and $(0.5; 0.5; 0)$ in terms of the number of self-study learners is negligible, for each school size. Since about half of all learners work on modules a second-degree teacher is qualified to teach, this was to be expected. This also suggests there is no obvious benefit to schedule quality when teacher qualifications are increased from a mix of first- and second-degree teachers to one of only first-degree teachers. On the other hand, introducing the third-degree qualification does result in a slight increase in the number of learners assigned to self-study, from on average 326 for $(0.5; 0.5; 0)$ to 358 for the medium school size, from 384 to 423 for the large school, and finally from 410 to 460 for the extra large school. The number of learners in self-study activities thus increases modestly by around 10%, suggesting it is indeed somewhat harder to schedule instruction activities when third-degree teachers make up 20% of the fixed-size teacher set T. This will not be a problem if a school hires third-degree teachers in addition to its existing first- and second-degree teachers, thus increasing the size of the teacher set.

We next investigate the effects of varying the number of third-degree teachers. With a slight abuse of notation, we write $(1 - q; 0; q)$ for the teacher qualification distribution, and vary q in $(0, 0.1, \ldots, 0.9, 1)$. For each value of q, σ, and each of the school sizes, we generate ten instances with $w = 50\%$, $\delta^- = 5$, and $\delta^+ = 30$, and regular classrooms. This results in 132 new experiments, and a total of 1,320 new instances. These are then solved, and the results are displayed in Fig. 7 as the percentage of learners in self-study activities as a function of q.

Fig. 7 suggests that for q up to one half the number of learners in self-study are not affected. For $q > 0.5$, the number of learners assigned to self-study activities quickly increases to 100% as insufficient qualified teachers are available for instruction. These results hold for different demand spreads σ and school sizes, and thus appear to be rather robust.

Part of the explanation for these results is that our experimental instances have twice as many teachers as (non-split) classrooms. Instruction activities can then mostly be planned as long as q remains below 0.5, since there is likely a teacher available: the classrooms are the bottleneck resource, not the teachers. By implication, this would require that the remaining first- and second-degree teachers spend most or all of their working time teaching instruction activities, which is not realistic since teachers also have administrative tasks, in addition to their teaching. Thus, q should not be set too large, and values around 0.1 or 0.2 might be preferred. Nonetheless, there appears to be ample room for third-degree teachers to supervise self-study assignments in PL.

6. Conclusions

In this paper we explore the hourly planning problem of personalised learning, where learners have varying demands for various course modules and need to be scheduled into feasible instruction and self-study activities. We propose an integer linear programming formulation capable of solving the problem to optimality. This formulation often does not solve within a few minutes, limiting its practical utility for hourly planning. An ALNS metaheuristic solution approach which solves the problem quickly was proposed next. The metaheuristic is validated on an extensive set of realistically-sized numerical experiments in the context of Dutch secondary education, generated based on expert insight. We find the metaheuristic performs very well, and on

average deviates only 1.6% in objective value from optimal solutions in our experiments. We numerically illustrate the difficulties of increased learner demand spread, and study the effect of several changes to the structure and staffing of secondary education to mitigate these difficulties. We show that changes to the structure of school buildings limits the number of self-study activities, and that a penalty term for learner assignments to self-study activities also offers an effective control on the number of self-study activities.

Several directions for future research are worthwhile to explore. Of practical relevance is a future paper that explores the modifications to the structure and staffing of existing schools that we investigated here in more detail, and over longer scheduling horizons. Further, exact approaches might be feasible for the HLAPP, even at scale. One way forward in this direction could proceed by formulating novel valid inequalities for our ILP, including symmetry breaking constraints on the classroom–teacher assignments. Alternatively, our ILP and solution approaches could be extended to cover longer scheduling horizons, in particular a single day or week.

CRediT authorship contribution statement

Niels A. Wouda: Conceptualization, Methodology, Software, Validation, Investigation, Writing – original draft. **Ayse Aslan:** Methodology, Writing – original draft, Writing – review & editing. **Iris F.A. Vis:** Conceptualization, Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Research data and code to replicate the results in this manuscript can be found at https://doi.org/10.34894/E2L6WC.

Acknowledgements

We thank Wim Kokx, president of the board of Openbare Scholengroep Vlaardingen Schiedam, for his expert insight and feedback. We would also like to thank two anonymous reviewers for their thoughtful comments and efforts towards improving this paper.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.cor.2022.106089.

References

Aslan, A., Bakir, I., Vis, I.F.A., 2020. A dynamic thompson sampling hyper-heuristic framework for learning activity planning in personalized learning. European J. Oper. Res. 286 (2), 673–688.

Asratian, A.S., De Werra, D., 2002. A generalized class–teacher model for some timetabling problems. European J. Oper. Res. 143 (3), 531–542.

Eiken, O., 2011. The Kunskapsskolan ("The Knowledge School"): A personalised approach to education. CELE Exch. Cent. Eff. Learn. Environ. 1, 1–4.

European Political Strategy Centre, 2019. 10 Trends transforming education as we know it. http://dx.doi.org/10.2872/800510, https://op.europa.eu/en/publication-detail/-/publication/227c6186-10d0-11ea-8c1f-01aa75ed71a1.

Johnes, J., 2015. Operational research in education. European J. Oper. Res. 243 (3), 683–696.

Kannan, A., van den Berg, G., Kuo, A., 2012. iSchedule to personalized learning. Interfaces 42 (5), 437–448.

Kiefer, A., Hartl, R.F., Schnell, A., 2017. Adaptive large neighborhood search for the curriculum-based course timetabling problem. Ann. Oper. Res. 252 (2), 255–282.

Kristiansen, S., Sørensen, M., Herold, M.B., Stidsen, T.R., 2013. The consultation timetabling problem at Danish high schools. J. Heuristics 19 (3), 465–495.

Kristiansen, S., Sørensen, M., Stidsen, T.R., 2011. Elective course planning. European J. Oper. Res. 215 (3), 713–720.

Kunskapsskolan, 2019. The KED network. http://www.kunskapsskolan.com/thekednetwork.

Lei, H., Liu, X., Laporte, G., Chen, Y., Chen, Y., 2018. An improved adaptive large neighborhood search algorithm for multiple agile satellites scheduling. Comput. Oper. Res. 100, 12–25.

Lindauer, M., Eggensperger, K., Feurer, M., Biedenkapp, A., Deng, D., Benjamins, C., Ruhkopf, T., Sass, R., Hutter, F., 2022. SMAC3: A versatile Bayesian optimization package for hyperparameter optimization. J. Mach. Learn. Res. 23 (54), 1–9, URL http://jmlr.org/papers/v23/21-0888.html.

Ministerie van Onderwijs, Cultuur en Wetenschap, 2020. Kamerbrief lerarenbeleid en onderwijsarbeidsmarkt 2020. https://www.rijksoverheid.nl/documenten/kamerstukken/2020/12/09/kamerbrief-lerarenbeleid-en-onderwijsarbeidsmarkt.

Pillay, N., 2014. A survey of school timetabling research. Ann. Oper. Res. 218 (1), 261–293.

Pisinger, D., Røpke, S., 2019. Large neighborhood search. In: Gendreau, M., Potvin, J.-Y. (Eds.), Handbook of Metaheuristics, 3rd ed. Springer International Publishing, pp. 99–127.

Post, G.F., Ahmadi, S., Daskalaki, S., Kingston, J.H., Kyngas, J., Nurmi, C., Ranson, D., 2012. An XML format for benchmarks in high school timetabling. Ann. Oper. Res. 194 (1), 385–397.

Roozbeh, I., Ozlen, M., Hearne, J.W., 2018. An adaptive large neighbourhood search for asset protection during escaped wildfires. Comput. Oper. Res. 97, 125–134.

Røpke, S., Pisinger, D., 2006. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transp. Sci. 40 (4), 455–472.

Santiago-Mozos, R., Salcedo-Sanz, S., DePrado-Cumplido, M., Bousoño-Calzón, C., 2005. A two-phase heuristic evolutionary algorithm for personalizing course timetables: a case study in a Spanish university. Comput. Oper. Res. 32 (7), 1761–1776.

Santini, A., Røpke, S., Hvattum, L.M., 2018. A comparison of acceptance criteria for the adaptive large neighbourhood search metaheuristic. J. Heuristics 24 (5), 783–815.

Sørensen, M., Kristiansen, S., Stidsen, T.R., 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, pp. 489—492.

Sørensen, M., Stidsen, T.R., 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, pp. 359–364.

Veenstra, M., Vis, I.F.A., 2016. School timetabling problem under disturbances. Comput. Ind. Eng. 95, 175–186.

Wen, M., Linde, E., Ropke, S., Mirchandani, P., Larsen, A., 2016. An improved adaptive large neighborhood search heuristic for the electric vehicle scheduling problem. Comput. Oper. Res. 76, 73–83.