10th International Conference of the Practice and Theory of Automated Timetabling
PATAT 2014, 26-29 August 2014, York, United Kingdom

A linear mixed-integer model for realistic examination timetabling problems

Lisa Katharina Bergmann · Kathrin Fischer · Sebastian Zurheide

Abstract An examination timetable has to satisfy a vast variety of requirements to be not only feasible, but also to be convenient to all parties involved. Many different aspects, as e. g. spreading of exams for students' convenience or fixing exams to certain days or rooms for teachers' convenience, have been discussed in the literature. However, there are no model formulations which take all aspects relevant for this work into account.

Therefore, in this work a new linear mixed-integer programming model for the exam timetabling problem is presented. The model uses a penalty-based goal programming approach to assure the construction of timetables which fulfill important requirements made by teachers, students and administrators. Based on this model, feasible solutions are derived by a standard solver and subsequently are further improved by a tabu-search procedure. The trade-off between different criteria is shown and some very promising results of the approach for a real-world data set are presented.

Keywords Examination timetabling · University timetabling · Linear programming · Mixed-integer programming

---

L. K. Bergmann
Institute for Operations Research and Information Systems
Hamburg University of Technology
Schwarzenbergstr. 95 D
21073 Hamburg
Germany
Tel.: +49-40-428784534
E-mail: katharina.bergmann@tuhh.de

K. Fischer
E-mail: kathrin.fischer@tuhh.de

S. Zurheide
E-mail: zurheide@tuhh.de

1 Introduction to exam timetabling at universities

The exam timetabling problem deals with the assignment of exams to rooms and time slots such that there is never more than one exam per student at a time. At universities, this combinatorial problem arises at the end of each term or semester. Usually there is a predefined time-span within each term, in which all exams have to be scheduled. For the assignment of exams to dates this overall examination time-span is split into several time slots (or exam periods) of equal length. The decision when and where to schedule an exam is quite challenging for the planner, and therefore tools which support the timetabling process are valuable.

One might assume that the exam timetable can be derived easily from the course timetable, as teachers might use the last session of their course for the respective examination. However, it is usually required that students do not have to take more than one exam per day, but as they can do more than one class per day, the course timetable cannot be used for the examinations. Moreover, there are many additional conditions to be fulfilled by an exam timetable, as will be explained below.

The course timetable is mainly determined by the curriculum of the respective degree program and is based on the assumption that every student participates in the courses scheduled in a term, and that he/she also passes the corresponding exams in the same term. In reality, students often take classes and exams in a different order. For example, due to lectures without compulsory attendance, illness during the examination time-span or failing of exams, the participation in courses and in the related exams is often rather independent from each other and hence the order in which exams are passed often does not follow the curriculum anymore. In addition, at German universities exams are usually offered in every term, even if the corresponding course is only taught every other term. Hence, about half of the exams that have to be scheduled are so-called resit exams which belong to courses that have not been taught in the current term. Moreover, it is usually required that the seating during an exam is more spacious than during a lecture (and thus a bigger room is required for the exam). Last but not least, there are usually many different electives between which students can choose, enhancing the danger of exam overlaps. All this makes the task of exam timetabling a very complex combinatorial problem.

A new approach for modeling the timetabling problem which is based on students' enrollments is presented in this work. Apart from the very basic requirements, e. g. that every student can only attend one exam per period, the linear mixed-integer model which is developed below contains several features in order to meet realistic demands of teachers and students, like the consideration of the work-load associated with an exam, the actual time-span between two exams or the distinction between exams from courses from the current term and resit exams. Additionally the changing availability of rooms, the need of splitting large exams over several rooms, the booking of external rooms for very specific exams and related to this also the preassignment of exams to dates or rooms is enabled. In order to implement all these requirements the model is based on a penalty based goal programming approach where deviations from soft requirements are penalized and hard requirements are integrated as constraints.

2 Literature review on examination timetabling

Several surveys on the topic state that the primary objective of exam timetable planning is to set up a conflict free schedule for every student (Carter (1986), Carter and Laporte (1996), Qu et al (2009)), i. e. a schedule in which no student has to take more than one exam at a time.

Resulting from this requirement, the most fundamental case of the exam timetabling problem is basically a graph coloring problem where each exam is represented by a node. If at least one student is enrolled for two exams, the corresponding nodes are connected by an undirected arc. The weights on the arcs equal the number of students enrolled in the two exams. The objective is then to find the minimal node coloring, where adjacent nodes do not have the same color, i. e. a schedule where no student is required to take more than one exam at a time. The colors can be identified with the available time slots. As the NP-hard coloring problem can be mapped polynomially onto the examination timetabling problem, the latter is also NP-hard (Garey and Johnson, 1979, pp. 13-14).

In the following a distinction between first and second order conflicts has to be made. A first order conflict arises if two exams are scheduled in the same period and there is at least one student enrolled in both exams. These conflicts have to be avoided by all means. Second order conflicts arise from exams that are "only" scheduled too close to each other, but not in the same period, and if there is at least one student enrolled for both exams.

In addition to the main requirement of being first order conflict-free, some additional hard constraints should be fulfilled by any timetable (Qu et al, 2009):

– Every exam has to be scheduled exactly once and
– Available capacities (rooms, invigilators, time) must not be exceeded.

If it is not possible to find a solution that satisfies the capacity requirements or the requirement of being conflict free, these constraints can be relaxed by adding dummy capacities or by adjusting the objective such that the number of first order conflicts is minimized (Carter and Laporte, 1996). However, this can lead to timetables that cannot be implemented in reality.

Depending on the individual demands of the university, there can be secondary objectives and requirements that can be included in the model as soft constraints. Table 1 lists some of the requirements that can be found in the literature and are often included in model formulations for realistic examination timetabling problems (more potential requirements can be found in Qu et al

Table 1 Overview on selected realistic examination timetabling requirements and authors considering them

Minimize the number of second order exam conflicts (i. e. spread the exams as much as possible over the entire examination time-span)	Müller (2013), Eley (2007), Burke and Newall (2004), Di Gaspero and Schaerf (2001), Carter et al (1994), Arani et al (1988), Laporte and Desroches (1984)
Preassign an exam to a specific time slot (or exclude this time slot)	Müller (2013), Carter et al (1994), Laporte and Desroches (1984)
Preassign an exam to a specific room or room type (or exclude this room/room type)	Müller (2013), Laporte and Desroches (1984)
Enable the consideration of preferred or required room types for some exams, e. g. rooms with large tables or computers	Carter et al (1994), Laporte and Desroches (1984)
Some rooms may not be available during the entire examination time-span	Müller (2013), Di Gaspero and Schaerf (2001), Laporte and Desroches (1984)
If big rooms are scarce it is possible to assign more than one room to an exam	Eley (2007)
Additionally the maximum number of rooms that an exam may be split into is limited	Müller (2013), Laporte and Desroches (1984)
Schedule more than one exam in a room (in the same time slot) if the number of available rooms is scarce	Eley (2007), Di Gaspero and Schaerf (2001), Laporte and Desroches (1984)
Penalize the scheduling of exams in certain time slots, e. g. avoid exams in the last day of the examination time-span	Müller (2013)
Enable the consideration of preferred time slots for some exams	Di Gaspero and Schaerf (2001), Carter et al (1994)

(2009)). Next to each aspect a few authors who consider these requirements in their articles are given.

The approaches by Müller (2013), Carter et al (1994) and Laporte and Desroches (1984) aim to set up realistic examination timetabling systems and consider many of the above named requirements, though none of them gives a complete mathematical formulation taking all aspects relevant for this work into account. Müller (2013) presents several benchmark data sets and an algorithm that resolves second order conflicts in several phases. Laporte and Desroches (1984) introduce an automatic timetabling procedure which includes the respective requirements from Table 1. Carter et al (1994) propose a scheduling system that is based on the article by Laporte and Desroches (1984) and implemented at the University of Toronto and at Carleton University. Carter et al (1996) also base their work on the article by Laporte and Desroches (1984) and carry out several experiments with algorithms that determine the optimal length of the examination time-span. These tests are conducted on unconstrained problems to determine the best strategy, which is then used to solve several real-life constrained problems. Carter et al (1996) do not explicitly state which requirements were considered for the constrained problems, therefore their work has not been included in Table 1.

The focus of the other publications listed above is primarily on the solution approach, and not on the formulation of a mathematical model. Eley (2007) uses different ant algorithms to solve the examination timetabling problem, Burke and Newall (2004) approach the problem through adaption of heuristic orderings, Di Gaspero and Schaerf (2001) present several variants of tabu-search based solution algorithms and Arani et al (1988) (and Arani and Lotfi (1989)) propose a three phase approach using a Lagrangian relaxation.

A common approach, which is also used by some of the authors mentioned above, to mathematically formulate the objectives and constraints listed in Table 1 is in the form of the quadratic assignment problem where the variables represent the assignment of exams to periods and sometimes also to rooms (Eley (2007), Burke and Newall (2004), Di Gaspero and Schaerf (2001), Arani and Lotfi (1989), Laporte and Desroches (1984)). For exams i and j being scheduled in period t and $t + a$, the corresponding constraints of the type "at most one exam per student within a certain time-span" can be formulated as quadratic constraints by the expression $c_{ij}x_{it}x_{j(t+a)} = 0$, with c_{ij} being the number of students that are enrolled for both exams i and j and x_{it} ($x_{j(t-a)}$) being equal to one if exam i (j) is scheduled in period t ($t + a$) and zero otherwise. Independent from the programming approach, the use of the conflict matrix c_{ij} can often be found in the literature, e. g. in Eley (2007), Burke and Newall (2004), Carter et al (1996) and Laporte and Desroches (1984).

For the topic of course timetabling, linear models are presented in the literature (e. g. Schimmelpfeng and Helber (2007), Van den Broek et al (2007), Dimopoulou and Miliotis (2001)). Schimmelpfeng and Helber (2007) model a course timetabling problem as a linear assignment problem. Using elements from goal programming, their model penalizes conflicting dates for courses and violations of room and teaching capacities. Instead of directly minimizing the number of students affected by a bad schedule, the model balances the students' work-load by minimizing the sum of penalties that apply in case of a bad schedule. Dimopoulou and Miliotis (2001) use a linear model to solve the course timetabling problem at a Greek university. Based on the solution of this model, an infeasible starting solution for the examination timetabling problem is generated, then modified into a feasible one by an algorithm that resolves the first order conflicts, and afterwards improved by rescheduling of exams.

Various other publications on the topic of examination timetabling can be found. An extensive literature review on examination timetabling is presented by Qu et al (2009). However, most publications focus on solution methods and use quadratic models whereas in this work, a linear approach is presented, to enable the use of standard solution approaches.

3 A new linear mixed-integer model for exam timetabling

Examination procedures may vary, e. g. depending on the educational institution, on the number of students, degree programs (timing, courses, choices), teachers, available facilities, technical support of the planner, and many more aspects. However, the following requirements and assumptions should match a typical German university.

The linear mixed-integer model that is introduced below considers all of the previously listed requirements, with a few exceptions regarding the room allocation. Experience from different German universities shows, that usually only one exam per room and period is scheduled. This is mainly due to the fact that German universities tend to have only very few large rooms, but enough small rooms, and hence it is usually possible to assign different exams to different (smaller) rooms.

Also the distinction of room types, apart from their capacity, is not explicitly included, but the model enables the exclusion of specific rooms for some exams (and therefore also the preassignment of a room to an exam). The model also includes a few additional requirements which are typical for German universities and were not taken into account by any of the above mentioned authors: If second order conflicts cannot be avoided, it is important to consider whether an exam belongs to a course from the current term, or to the previous term. Exams for courses from the current term are considered to be more important, as universities want to encourage the students to follow the curriculum as closely as possible. Hence, second order conflicts are to be avoided especially if both exams belong to courses from the current term.

This is done by using a goal programming based approach, such that the model penalizes three things: the occurrence of second order conflicts, the splitting of an exam over several rooms and the scheduling of exams in undesirable periods. The approach takes into account the number of students involved in second order conflicts, and the exams' work-loads. A similar approach for course timetabling can be found in Schimmelpfeng and Helber (2007).

3.1 General conditions and assumptions

It is assumed that students enroll for the exams they want to take in an examination period before the examination timetable is created, and that there are no exams without enrollments. In addition to the given enrollments, the examination time-span is split into periods and the total number of available periods within the overall examination time-span is predetermined. However, the number of available periods is varied in the computational study presented below to examine the effect of different examination time-spans. The periods are of equal duration, which is given in hours. E. g. an exam day that starts at 8 a.m. and ends at 6 p.m. can be represented by a period length of 10 hours and a total number of periods that equals exactly the number of days in the examination time-span (i. e. one period per day). Alternatively it is possible to represent each day by two periods, each being 4 hours long, or by any other number of periods per day. It only has to be made sure that the periods are of equal length, and of course there will be no feasible solution, if the duration of any exam exceeds the period duration. Hence, the exams' durations also have to be given in hours. This also allows to schedule more than one exam in the same room and period, if the sum of the exams' durations does not exceed the duration of a period. I. e. these exams are scheduled *subsequently* in the same room and within the same period. The exact schedule and order of exams in a room can easily be set up manually.

For the students it is very important that the work-loads associated with the exams are taken into account, especially when second order conflicts cannot be avoided. Therefore, in addition to the enrollments and duration of each exam, the ECTS points (European Credit Transfer and Accumulation System) for every exam have to be considered by the model.

In realistic timetabling situations, the room availabilities may not be the same for every period. E. g. conferences or other events might take place during part of the examination time-span, and hence certain rooms are not available at certain times. Moreover, it is necessary to enable the model to predetermine the period for an exam or exclude some periods for certain exams. The same has to be possible for the room allocation: e. g. for a very large examination, an external room might be booked, but other smaller exams should not take place in that room. To allow a reasonable assignment of rooms and to limit the number of invigilators needed, the maximum number of rooms into which an exam may be split has to be limited.

Furthermore the model allows to specify a "room allocation enrollment limit". If this is done, only exams that exceed the limit have to be scheduled with room. Smaller exams only get a period assigned. As pointed out above, at many German universities, small rooms (e. g. for seminars or group work) are available in abundance, such that the manual assignment of appropriate rooms to small exams is easily possible. This can lead to a considerable reduction of model size.

Finally it should be possible to avoid scheduling of exams in certain periods if this is feasible. E. g. students as well as teachers usually prefer not to have any exams in the very first or last periods of the overall examination time-span which usually starts right after the end of the term. If also weekends can be used for examinations, their use should also be avoided whenever possible.

3.2 Sets and parameters used in the model

In the following, the sets, indices and parameters which are needed to include the stated requirements and assumptions in the model are introduced. I is the set of all exams, and the indices for exams are i and j. Exams may be scheduled in a room r out of all rooms R. The total number of available exam periods is given by P with p being the index for periods such that $p \in \{1..P\}$.

To balance the students' work-load, they should not be required to take more than one exam within a certain number of periods. The length (in periods) of this time-span is defined by A, indexed by a, $a \in \{1..A\}$. If an exam is scheduled in period p, a penalty is applied if another exam is scheduled in a period $p+1$ to $p+A$, if there is at least one student enrolled in both exams.

To enable the distinction between exams from the current term and resit exams from the previous term, several sets of ordered tuples are introduced. Each tuple contains two exams that have a conflict potential, i. e. there is at least one student enrolled in two exams, i and j. These tuples of conflicting exams are generated in advance, based on the students' enrollment information. In the following sections and chapters, conflicting exams will be denoted by i' and i''.

The set I^T contains all exam pairs that have a conflict potential, i. e. all exam pairs where at least one student enrolled for both exams. I^{TC} contains only tuples of exams that both belong to courses from the current semester, while I^{TP} contains only tuples where at least one exam belongs to a course from the previous semester, such that $I^T = I^{TC} \cup I^{TP}$. All sets of tuples are indexed by (i', i'').

The parameter E_i gives the number of students that are enrolled for an exam $i \in I$ while the number of students that are enrolled for two exams i' and i'' (thus with conflict potential) is given by the parameter $C_{(i',i'')}$.

To reduce the problem size, the original set of all exams I is also complemented by two subsets: I^Z and I^W. I^Z is the set of all "big" exams with E^Z or more enrollments. Only exams in this set have to be scheduled with a room. As E^Z is a parameter of the model, it is of course possible to set its value to 1; then all exams will be scheduled with a room. Analogously I^W is the set of all exams with special time requirements. If no time requirement exists, the exam can be scheduled in any period. But if e. g. a teacher is absent during specific periods, his or her exam should not be scheduled in these periods (see also O_{ip} below).

The individual work-load or severity for each exam is represented by the parameter S_i. It takes a value between 1 and 5, $S_i \in \{1..5\}$. E. g. if i is a very easy exam, with a low work-load (and thus few ECTS points), S_i equals 1, but if i is a very difficult exam, with high work-load (and many ECTS-points), S_i equals 5. The difficulty is combined with $C_{(i',i'')}$ to determine the parameter $B_{(i',i'')}$, which gives the "badness" that occurs if exams i' and i'' are scheduled too close to one another. It is obtained by multiplying $C_{(i',i'')}$ with the exams' severities. Based on discussions and a university internal survey with students it is assumed that they prefer taking a difficult exam first and then an easier one, instead of the other way around. Hence, the "badness" value of a tuple is doubled if i'' has a higher work-load than i', i. e. $S_{i'} > S_{i''} \Rightarrow B_{(i',i'')} = 2C_{(i',i'')}S_{i'}S_{i''}$. Therefore, the elements of B are not symmetric (unlike $C_{(i',i'')}$).

The duration of the individual periods is given in hours and denoted by H. If H is set to a relatively large number (e. g. 8 hours), a high occupancy of each room can only be achieved by allowing the scheduling of more than one exam in a room. Therefore, the duration of each exam needs to be considered, to make sure that several exams can take place consecutively in the same room. This individual duration D_i gives the hours needed for each exam i, including time for preparation and follow up.

K_{rp} is a $(m \times P)$ matrix, with m being the total number of available rooms, and P the total number of available periods. The elements of K_{rp} are positive integer or zero and give the number of seats that are available in room r during period p. If a room is not available in a specific period, the value for the corresponding K_{rp} is set to zero. The time requirements for every exam are specified by the binary parameter O_{ip}. It equals 1 if exam i may be scheduled in period p and 0 otherwise. If a special time requirement for an exam i exists ($\sum_{p\in\{1..P\}} O_{ip} < P$), the exam will be added to the set of exams with time requirements, I^W, as mentioned above. Similar to this Q_{ir} is a binary parameter which specifies the room requirements for every exam: It equals 1 if exam i may be scheduled in room r and 0 otherwise. The parameter F limits the number of rooms into which an exam may be split.

There are four penalty factors to enable a weighting of the different terms of the objective function. (Note that the superscripted number is an index to distinguish between the different parameters and not a mathematical exponent.) N_a^1 gives the penalty that applies if conflicting exams are scheduled within $A+1$ periods, with $a \in \{1..A\}$. It enables the model to penalize second order conflicts with respect to the actual distance of time, a, of the corresponding exams (similar to Eley (2007) or Carter et al (1996)). N^2 determines the impact of second order conflicts of two exams from current courses compared to conflicts with an exam from a previous term course. N^3 defines the level of the penalty for splitting an exam into several rooms. Finally, N_p^4 penalizes the scheduling of exams in undesirable periods, e. g. all periods representing a Saturday. These factors can of course also be used to favor certain schedules (instead of penalizing them) if the values are set appropriately (i. e. if the penalty values are set between 0 and 1).

3.3 Decision and deviational variables

The timetabling model presented below comprises two types of decision variables. There are two classical decision variables, y_{irp} and x_{ip}, and two so called deviational variables, $u_{(i',i'')a}$ and v_{ip}.

The variable y_{irp} gives the information in which room and period an exam is scheduled. Hence, it is equal to 1 if exam i is scheduled in room r in period p, and otherwise it is zero. The second decision variable, x_{ip}, only gives the information in which period an exam i is scheduled. x_{ip} is equal to 1 if exam i is scheduled in period p and equal to 0 otherwise. Due to this separate allocation of periods and rooms it is possible to ensure that every exam is scheduled exactly once, but to allow several rooms to be assigned to one exam.

The first deviational variable $u_{(i',i'')a}$ is binary and indicates whether a student has to sit two exams i' and i'' within $A+1$ periods or not. If exam i' is scheduled in period p and exam i'' is scheduled in period $p+a$ then $u_{(i',i'')a}$

takes the value 1. If there is no student enrolled in both exams or there are more than A periods in between the exams, $u_{(i',i'')a}$ takes the value 0. The index a contains the information how many periods there are in between the periods in which the two exams are scheduled.

The number of additional rooms that are assigned to exam i scheduled in period p is given by v_{ip}. E. g. if exam i is scheduled in period p and assigned to three rooms, then $v_{ip} = 2$. On the other hand, if $v_{ip} = 0$, exam i is either not scheduled in period p or it is assigned to just one room. These two decision variables are thus called positive deviational variables (Jones and Tamiz, 2010, pp. 4-5, 20-22).

3.4 Model formulation

To provide a better overview of the notation used, the following lists contain all sets, indices, parameters, decision and deviational variables.

Sets:

I	Set of all exams
I^W	Set of exams with time specification
I^Z	Set of exams with E^Z or more enrollments
I^T	Set of ordered tuples of all exam pairs with conflict potential, $I^T = I^{TC} \cup I^{TP}$
I^{TC}	Set of ordered tuples of exams with conflict potential that belong to courses from the current term
I^{TP}	Set of ordered tuples of exams with conflict potential, where at least one exam is a resit exam
R	Set of rooms

Indices:

1–4	Naming indices for penalty factors
a	Index for "time lags" between two exams, $a \in \{1..A\}$
i, j	Indices for exams, $i, j \in I, I^Z$ or I^W
(i', i'')	Tuple of indices for exam pairs with conflict potential, $(i', i'') \in I^T, I^{TC}$ or I^{TP}
p	Index for periods, $p \in \{1..P\}$
r	Index for rooms, $r \in R$

Parameters:

A	Number of consecutive periods in which no student should have to write more than one exam
$B_{(i',i'')}$	Badness for exams of tuple (i', i'') being scheduled too close
D_i	Duration of exam i
E_i	Number of students enrolled for exam i

E^Z	Limit of enrollments, such that all exams with E^Z or more enrollments have to be scheduled with room
F	Limit on rooms, an exam may be split into
H	Length of each period in hours
K_{rp}	Seating capacity of room r in period p
N_a^1	Penalty factor that applies if exams of a tuple are scheduled within $a+1$ periods
N^2	Additional penalty factor that only applies for tuples $(i', i'') \in I^{TC}$
N^3	Penalty factor that applies if an exam is split into several rooms
N_p^4	Penalty factor that depends on the period in which an exam is scheduled
O_{ip}	Time specification that indicates if institutes/teachers prefer exam i to be scheduled in period p or not
P	Total number of available periods, being H hours long each
Q_{ir}	Room specification that indicates if exam i may be scheduled in room r or not
S_i	Work-load (severity) of exam i

Decision variables:

x_{ip}	Equal to 1 if exam i is scheduled in period p, and 0 otherwise
y_{irp}	Equal to 1 if exam i is scheduled in room r in period p, and 0 otherwise

Deviational variables:

$u_{(i',i'')a}$	Equal to 1 if exams of tuple (i', i'') are scheduled within $a+1$ periods, and 0 otherwise
v_{ip}	Equal to the number of additional rooms occupied by exam i in period p, and 0 otherwise

The following model formally states a realistic exam timetabling problem, taking all above mentioned aspects into account, and using the notation presented.

Objective function:

$$\min \sum_{a \in \{1..A\}} N_a^1 \left(\sum_{(i',i'') \in I^{TC}} N^2 B_{(i',i'')} u_{(i',i'')a} + \sum_{(i',i'') \in I^{TP}} B_{(i',i'')} u_{(i',i'')a} \right) + \sum_{i \in I} \sum_{p \in \{1..P\}} N^3 v_{ip} + \sum_{i \in I} \sum_{p \in \{1..P\}} N_p^4 S_i x_{ip} \tag{1}$$

Subject to:

$$\sum_{p\in\{1..P\}} x_{ip} = 1 \qquad \forall i \in I \tag{2}$$

$$x_{ip} \le O_{ip} \qquad \forall i \in I^W, \forall p \in \{1..P\} \tag{3}$$

$$x_{i'p} + x_{i''p} \le 1 \qquad \forall (i', i'') \in I^T, \forall p \in \{1..P\} \tag{4}$$

$$x_{i'p} + x_{i''(p+a)} - u_{(i',i'')a} \le 1 \qquad \forall (i', i'') \in I^T, \forall p \in \{1..P\}, \forall a \in \{1..A\} \tag{5}$$

$$\sum_{p\in\{1..P\}} y_{irp} \le Q_{ir} \qquad \forall i \in I^Z, \forall r \in R \tag{6}$$

$$\sum_{r\in R} y_{irp} - v_{ip} = x_{ip} \qquad \forall i \in I^Z, \forall p \in \{1..P\} \tag{7}$$

$$\sum_{r\in R} y_{irp} \le F \qquad \forall i \in I^Z, \forall p \in \{1..P\} \tag{8}$$

$$\sum_{r\in R} K_{rp} y_{irp} \ge E_i x_{ip} \qquad \forall i \in I^Z, \forall p \in \{1..P\} \tag{9}$$

$$\sum_{i\in I^Z} D_i y_{irp} \le H \qquad \forall r \in R, \forall p \in \{1..P\} \tag{10}$$

$$y_{irp} \le x_{ip} \qquad \forall i \in I^Z, \forall r \in R, \forall p \in \{1..P\} \tag{11}$$

$$x_{ip} = 0 \qquad \forall p \in \{P+1..P+A\}, \forall i \in I \tag{12}$$

$$x_{ip} \in \{0,1\} \qquad \forall i \in I, \forall p \in \{1..P\} \tag{13}$$

$$y_{irp} \in \{0,1\} \qquad \forall i \in I^Z, \forall r \in R, \forall p \in \{1..P\} \tag{14}$$

$$u_{(i',i'')a} \in \{0,1\} \qquad \forall (i', i'') \in I^T, \forall a \in \{1..A\} \tag{15}$$

$$v_{ip} \ge 0 \qquad \forall i \in I^Z, \forall p \in \{1..P\}. \tag{16}$$

Constraints (2) ensure that every exam is scheduled exactly once. Equations (3) take the time requirements of institutes and teachers into account. They also allow to predetermine the period p in which a certain exam i has to be scheduled, by setting O_{ip} to 1 for only this specific period p.

As represented in the constraints (4), no student can take two exams at the same time, hence, first order conflicts cannot occur in a feasible solution. Constraints (5) relate to exams being scheduled within $A + 1$ periods. As both equations only apply to exams that have a conflict potential, they only have to hold for the tuples of the set I^T. In (5) the first set of deviational variables, $u_{(i',i'')a}$, is used: Whenever there is at least one student enrolled in both exams i' and i'' and these exams are scheduled within $A+1$ consecutive periods, $u_{(i',i'')a}$ takes the value 1.

The next six groups of constraints relate to exams that have to be scheduled with a room, as they have E^Z or more participants; these are all exams $i \in I^Z$. Constraints (6) determine the room specifications for each exam, similar to the time specification in (3). Only if Q_{ir} is equal to 1, the exam i may be scheduled in room r. This allows to predetermine the location(s) for exams, release rooms only for selected exams and to exclude specific exams from a certain room.

In some cases it might be necessary to split up the participants of one exam over several rooms. Equations (7) enable this, using the deviational variables v_{ip}, which equal the additional number of rooms needed for an exam. Hence their values should be minimized, as it is preferred not to split up exams. The maximum number of rooms in which an exam might take place is determined in constraints (8). Restrictions (9) make sure that the rooms which are assigned to an exam have enough seats for all enrolled students.

Constraints (10) allow several exams to be scheduled in the same room and period, as long as the sum of their durations does not exceed one period. Constraints (11) restrict the assignment of rooms to the period in which the corresponding exam is scheduled. Finally, constraints (12) to (16) declare the decision and deviational variables' domains.

The objective function (1) aims to find a schedule that keeps the work-load for each student balanced, but also meets the university's resources. This is done by minimizing the deviations from the following three targets: First, every student should take at most one exam (e. g. exam i') within $A + 1$ periods. For every additional exam within this time range (e. g. exam i''), the deviational variable $u_{(i',i'')a}$ takes the value 1. The index a indicates within how many periods these two exams are scheduled and is one of three factors that determine the influence of a second order conflict on the objective function value. If the values of N^1_a are inverse proportional to a it is ensured that the closer the two corresponding exams are scheduled, the larger the contribution of a conflict to the objective function value will be.

The second influencing factor is the so-called badness $B_{(i',i'')}$, a combination of the number of affected students and the work-load of each exam, as was explained in subsection 3.2. The last influencing factor is the information whether the tuple (i', i'') contains an exam from the previous semester or not. $B_{(i',i'')}$ is multiplied by N^2 if the tuple (i', i'') contains only exams from courses from the current semester.

Second, every exam should be scheduled in only one room. If, however, the students enrolled for an exam do not fit into the biggest room available, it is possible to split the exam over several rooms. For an exam i, scheduled in period p, v_{ip} indicates how many additional rooms are required. The factor N^3 enables an adequate weighting of this target and penalizes the use of additional rooms.

The last target considered by the objective function (1) is the general choice of favorable periods or, more explicitly, the choice of inconvenient periods. Depending on the values of N^4_p, scheduling exams e. g. at the beginning and at the end of the overall exam period is penalized. Especially difficult exams should not be scheduled right in the first periods to grant the students enough time for preparation. For this reason, the penalty is multiplied by the work-load S_i of the corresponding exam.

For a better understanding of the objective function, a small example of an exam schedule is presented in the following. Table 2 shows a timetable with

Table 2 Example schedule

	Periods (p)				
	1	2	3	4	5
Room 1	i_D	i_E			i_F
Room 2	i_D		i_G	i_H	
Example Student	i_D	i_E			i_F
N_p^4	50	30	1	40	80

five periods, two rooms and five exams (named i_D to i_H). Furthermore, the enrollments of an example student and the values of the parameter N_p^4 can be found in this table.

The example student is enrolled for the exams i_D, i_E and i_F. It is assumed that 20 other students have the same exam conflicts. The degrees of difficulty for these three exams are as follows: $S_D = 3$, $S_E = 5$ and $S_F = 1$. The exams i_D and i_E are from the current semester, and the exam i_F belongs to a course from the previous semester. The values of the remaining parameters are $A = 3$, $N_1^1 = 100$, $N_2^1 = 10$, $N_3^1 = 1$, $N^2 = 2$ and $N^3 = 1000$.

The badness of the exam tuples needs to be determined, based on the formula presented in subsection 3.2, i. e. $B_{(i',i'')} = [2]\, C_{(i',i'')} S_{i'} S_{i''}$. Hence, $B_{(D,E)}$ is equal to $2 \cdot 20 \cdot 3 \cdot 5 = 600$, $B_{(D,F)}$ is equal to $20 \cdot 3 \cdot 1 = 60$ and $B_{(E,F)}$ is equal to $20 \cdot 5 \cdot 1 = 100$. Here, the first exam tuple has a high value for the parameter $B_{(D,E)}$, because the conflicting exams have a high work-load, and the exam with the lower work-load is scheduled first. The following calculation shows how the objective function value for this small example is computed:

$$\begin{array}{ccccccccccc} N_1^1 & & N^2 & & B_{(D,E)} & & u_{(D,E)1} & & & & \\ \downarrow & & \downarrow & & \downarrow & & \downarrow & & & & \\ (100 & \cdot & 2 & \cdot & 600 & \cdot & 1 & + & & & \\ & & & & & & & & & & \\ N_3^1 & & B_{(E,F)} & & u_{(E,F)3} & & N^3 & & v_{D1} & & \\ \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \\ 1 & \cdot & 100 & \cdot & 1) & + & (1000 & \cdot & 1) & + & \\ & & & & & & & & & & \\ N_1^4 & & S_D & & x_{D1} & & & & & & \\ \downarrow & & \downarrow & & \downarrow & & & & & & \\ (50 & \cdot & 3 & \cdot & 1 & + & & & & & \\ & & & & & & & & & & \\ N_2^4 & & S_E & & x_{E2} & & & & & & \\ \downarrow & & \downarrow & & \downarrow & & & & & & \\ 30 & \cdot & 5 & \cdot & 1 & + & \ldots) & & & & \end{array}$$

In the first part of the calculation, the facts that the exams i_D and i_E are planned without a period inbetween ($a = 1$) and that there is only a gap of two periods between the exams i_E and i_F ($a = 3$) are penalized. The time gap between the exams i_D and i_F is big enough so that it is not penalized. The first exam conflict (i_D, i_E) accounts for 120 000 units of penalty costs, because the exams are scheduled in consecutive periods, both exams are from the current semester and the badness value for this tuple is very high. The conflict between the exams i_E and i_F results only in penalty costs of 100, because the exam i_F is for a course from the previous semester, there is a time gap of two days between the exams and the badness value is low.

The second part of the calculation represents the penalties for splitting the exam i_D over two rooms. Each additional room is penalized with a penalty value of 1000, hence the contribution to the objective function value is 1000 in this case. The last part of the objective function adds penalty costs for each exam depending on the scheduled period. For example, penalty costs of 150 are added for scheduling the exam i_D in the first period. The other exams are penalized accordingly.

4 Experimental results

The easiest way to derive a feasible solution for the model presented above is using a standard solver, e. g. Gurobi Solver or IBM ILOG CPLEX Optimization Studio. For the test case given below, Gurobi constructed feasible solutions for different numbers of available periods. Due to the size of the test case, which has been chosen to meet the size of realistic instances, and the complexity of the problem these timetables were not very good (in terms of the objective function values), even after two days of run-time. Therefore the solutions found by the solver were further improved by a tabu-search based heuristic.

As student numbers are relevant for the respective contribution to the objective function, it shows that second order conflicts have a major impact on the objective function value. Therefore, the heuristic focuses on resolving these conflicts based on a feasible start solution which is constructed by Gurobi and handed over in the beginning.

To improve the provided timetable there are four possible moves that can be performed by the heuristic. The first move tries to move a single exam to a new period, the second tries to exchange the periods of two exams and the third move tries to exchange the periods of three exams. These moves are only performed if they improve the timetable. Following the idea of tabu-search the fourth move may also deteriorate the schedule, by exchanging all the exams of two periods. By this the heuristic can escape from local optima.

As a test case, the original examination data from the Hamburg University of Technology (TUHH) of the winter term 2012/2013 is used. In that term there were 243 exams to be scheduled, and a total of 23 317 enrollments that yield 7132 tuples of exams with conflict potential. The durations of the examinations and the capacities of the rooms are based on the original data of the university. The values for the work-load of each exam were set according to the ECTS points of the corresponding courses. At this university the exams are scheduled in the lecture-free time at the end of each term. The relevant time-span amounted to 39 days (= number of available periods P)

in the corresponding term. It would be advantageous if this time-span could be shortened, to provide students with more time for internships, laboratory work or vacation. It turned out that the minimum number of days for which a feasible solution could be generated is 24. Hence, the number of periods P was varied from 24 to 39 to study the effect of the length of the examination time-span. For all cases the number of consecutive periods, in which no student should have to write more than one exam, was set to $A = 3$.

Values for the penalty parameters were defined as follows: The factors that penalize conflicts according to the actual number of periods in between the two corresponding exams are $N^1_a = 100$ (for $a = 1$), 10 (for $a = 2$) and 1 (for $a = 3$). The penalty that applies for conflicts of exams from the current term is $N^2 = 2$. For each additional room that an exam is split to the penalty is $N^3 = 1000$ and finally, to guide the spreading of exams the last penalty-vector is $N^4_p = \{60, 50, 40, 20, 5, 5, 5, 1, \ldots 1, 10, 10, 20, 20, 30, 60, 80, 100, 140, 180, 220, 260\}$ where all values are set to 1 for periods greater 7 and smaller $(P - 11)$.

Feasible solutions for the different time-spans were generated by the Gurobi Solver (version 5.5) on a computer with two 2.27 GHz Intel Xeon quad core processors and 24 GB RAM. The model for 39 periods consists of 1 140 612 rows, 237 496 columns and 3 888 846 non-zeros in the coefficient matrix. As the resulting solutions after two days of optimization time still showed a huge integrality gap, they were afterwards improved by the tabu-search based heuristic.

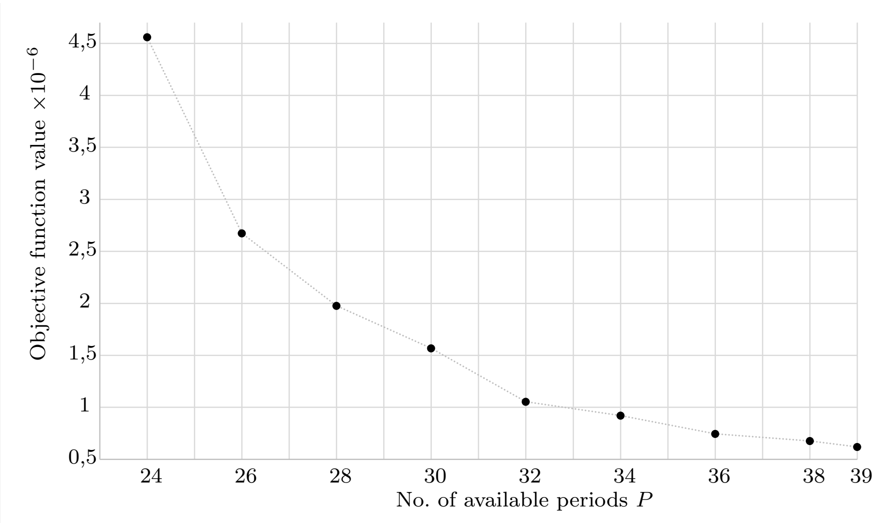

Fig. 1 Improved solutions for different numbers of available periods P

Figure 1 shows the resulting objective function values for different numbers of available periods P. The downward trend meets the expectation that the incurred penalties decrease with an increase of the total number of available

Table 3 Number of second-order conflicts for selected improved timetables

Number of available periods P	24 periods		28 periods		39 periods	
Total number of conflicts	762		640		379	
- affecting 10 or less students	602	(79%)	512	(80%)	327	(86%)
- affecting only one student	215	(28%)	196	(31%)	126	(33%)

periods, as exams with conflict potential can more easily be scheduled with a greater distance of time. In fact a closer look at the solutions shows that (e. g. due to the chosen values of the penalty factors) the main contribution to the objective function value results from second order conflicts: For the improved 24-periods-solution there are 762 second order exam conflicts (out of the 7132 exam pairs with conflict potential), but only seven exams (out of 243) had to be split into several rooms (actually exactly two rooms per exam). For the 28-periods-solution the improved timetable shows 640 second order conflicts and 11 exams were split into two rooms. For the 39-periods-solution the number of second order conflicts reduces to 379, and also 11 exams were split into two rooms.

For these three solutions, the number of conflicts can also be found in Table 3, which shows that the occurring second order conflicts mostly affect very few students. Moreover, a closer look at the solutions revealed that only very few of these conflicts involve two exams from courses from the current term.

The histogram in Figure 2 shows for the initial and improved 39-periods solution the number of conflicts that result from exams being scheduled too closely to each other. The horizontal axis gives the range in which the penalty costs induced by the conflicts lie, while the vertical axis gives the number of secondary conflicts per range. The total number of conflicts is reduced from initially 532 to 379. The figure shows that especially the number of conflicts with large penalties, i. e. the number of conflicts that either affect many students or concern difficult exams (or both), is significantly reduced by the heuristic procedure.

The original examination timetable, as it was (manually) generated and executed at the TUHH in the winter term 2012/2013 is not displayed in Figure 1. With respect to the model formulation in subsection 3.4, the original timetable was infeasible: There were 55 first order conflicts, i. e. there were 55 exam pairs scheduled on the same day, although there were students enrolled in both corresponding exams. Additionally, there were a few minor room size mismatches, i. e. some exams were scheduled in too small rooms. In practice this is not a problem, as there are usually a few students who do not attend every exam they are enrolled for, such that a slightly smaller room suffices. Ignoring all infeasibilities, the executed solution results in an objective function value of 736 882, but, due to the modifications that were necessary to resolve the infeasibilities, it is not advisable to compare this timetable to those constructed by the approach suggested in this work.

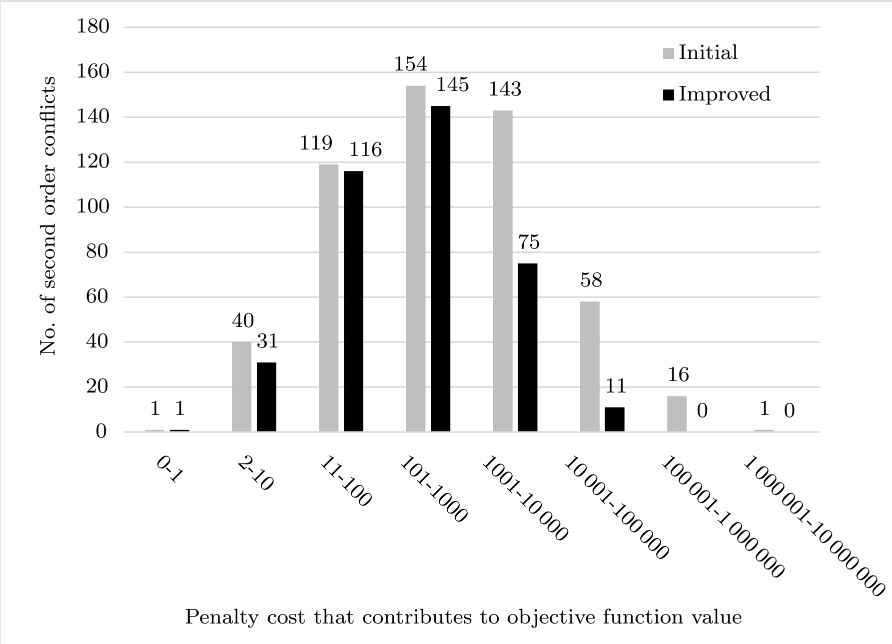

Fig. 2 Number of conflicts for initial and improved 39-periods-timetables

The results show that a compromise has to be found between the length of the examination period and the number of second order conflicts, as there is a trade-off between the two. Of course, which compromise is best may differ and depends largely on the preferences of the decision makers at the respective university. E. g. from Figure 1 it can be concluded, that in the specific case under study the shortening of the examination time-span by one week would lead to a still acceptable level of conflicts, which is only slightly higher than the one for the 39-period solution; so this might be a good compromise for this situation.

5 Summary and outlook

In this work, a linear mixed-integer model for the examination timetabling problem is presented. This model includes not only the fundamental constraints that have to be fulfilled, but also possibilities to set certain exams in specific periods or rooms. Furthermore, conflicts of exams are weighted based on the work-load, the time-span between exams and the term in which the respective course is taught. Based on students' enrollments and the resulting conflicts the model is aimed at finding the feasible solution which minimizes these conflicts, avoids the splitting of exams over rooms and the assignment of exams to inconvenient periods.

Several feasible timetables that were constructed by the Gurobi solver and improved by a tabu-search based heuristic demonstrate that the approach presented in this work can contribute substantially to the overall contentment of students and university teachers, as it is able to reduce the number of conflicts considerably compared to solutions which are constructed manually. The generated timetables could be used in practice at the TUHH, if the exam scheduling could be based on the students' enrollments. However, due to a different organizational approach this is currently not the case.

Concerning the constraints and targets of the model, there are still many issues that might be included in the future, e. g. the introduction of room types to satisfy special requirements of some exams (e. g. if computer workstations or tables for drawing are needed) or the assignment of exact start times for exams (not only the period) and different treatment of exams from compulsory and elective courses. Additionally, further analyses on the penalty parameters of the model or with respect to solution strategies might be carried out. An exact solution procedure like column generation might be able to solve realistic instances to optimality. These aspects are left for future research.

References

Arani T, Lotfi V (1989) A Three Phased Approach To Final Exam Scheduling. In: IIE Transactions, vol 21 (1), Taylor & Francis, pp 86–96

Arani T, Karwan M, Lotfi V (1988) A Lagrangian relaxation approach to solve the second phase of the exam scheduling problem. In: European Journal of Operational Research, vol 34, Elsevier Science B.V., pp 372–383

Van den Broek J, Hurkens C, Woeginger G (2007) Timetabling Problems at the TU Eindhoven. In: Burke EK, Rudová H (eds) PATAT 2006. LNCS, vol 3867, Springer, pp 210–227

Burke EK, Newall JP (2004) Solving examination timetabling problems through adaption of heuristic orderings. In: Annals of Operational Research, vol 129, Kluwer Academic Publishers, pp 107–134

Carter MW (1986) A Survey of Practical Applications of Examination Timetabling Algorithms. In: OR Practice, vol 34, Elsevier Science B.V., pp 193–202

Carter MW, Laporte G (1996) Recent Developments in Practical Examination Timetabling. In: Burke EK, Ross EK (eds) PATAT 1995. LNCS, vol 1153, Springer, pp 1–21

Carter MW, Laporte G, Chinneck JW (1994) A General Examination Scheduling System. In: Interfaces 24, pp 109–120

Carter MW, Laporte G, Lee SY (1996) Examination Timetabling: Algorithmic Strategies and Applications. In: Journal of the Operations Research Society 47, pp 373–383

Di Gaspero L, Schaerf A (2001) Tabu Search Techniques for Examination Timetabling. In: Burke EK, Erben W (eds) PATAT 2000. LNCS, vol 2079, Springer, pp 104–117

Dimopoulou M, Miliotis P (2001) Implementation of a university course and timetabling system. In: European Journal of Operational Research, vol 130, Elsevier Science B.V., pp 202–213

Eley M (2007) Ant Algorithms for the Examination Timetabling Problem. In: Burke EK, Rudová H (eds) PATAT 2006. LNCS, vol 3867, Springer, pp 364–382

Garey MR, Johnson DS (1979) Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, San Francisco

Jones D, Tamiz M (2010) Practical Goal Programming, 1st edn. Springer

Laporte G, Desroches S (1984) Examination Timetabling by Computer. In: Computers and Operations Research, vol 11, No. 4, pp 351–360

Müller T (2013) Real-life examination timetabling. In: Proceedings of the 6th Multidisciplinary International Scheduling Conference

Qu R, Burke EK, McCollum B, Merlot LTG, Lee S (2009) A survey of search methodologies and automated system development for examination timetabling. In: Journal of Scheduling, vol 12, Springer, Heidelberg, pp 55–89

Schimmelpfeng K, Helber S (2007) Application of a real-world university-course timetabling model solved by integer programming. In: OR Spectrum, vol 29, Springer, pp 783–803