Fairness in Examination Timetabling: Student Preferences and Extended Formulations

Ahmad Muklason[a,c,*], Andrew J. Parkes[a], Ender Özcan[a], Barry McCollum[b], Paul McMullan[b]

[a] *ASAP Group, School of Computer Science, University of Nottingham, Nottingham, NG8 1BB, UK*
[b] *School of Computer Science, Queen's University, Belfast, BT7 1NN, UK*
[c] *Department of Information Systems, Faculty of Information Technology, Institut Teknologi Sepuluh Nopember, Jl. Raya ITS, Kampus ITS Sukolilo, Surabaya, 60111, Indonesia*

Abstract

Variations of the examination timetabling problem have been investigated by the research community for more than two decades. The common characteristic between all problems is the fact that the definitions and data sets used all originate from actual educational institutions, particularly universities, including specific examination criteria and the students involved. Although much has been achieved and published on the state-of-the-art problem modelling and optimisation, a lack of attention has been focussed on the students involved in the process. This work presents and utilises the results of an extensive survey seeking student preferences with regard to their individual examination timetables, with the aim of producing solutions which satisfy these preferences while still also satisfying all existing benchmark considerations. The study reveals one of the main concerns relates to fairness within the students cohort; i.e. a student considers fairness with respect to the examination timetables of their immediate peers, as highly important. Considerations such as providing an equitable distribution of preparation time between all student cohort examinations, not just a majority, are used to form a measure of fairness. In order to satisfy this requirement, we propose an extension to the state-of-the-art examination timetabling problem models widely used in the scientific literature. Fairness is introduced as a new objective in addition to the standard objectives, creating a multi-objective problem. Several real-world examination data models are extended and the benchmarks for each are used in experimentation to determine the effectiveness of a multi-stage multi-objective approach based on weighted Tchebyceff scalarisation in improving fairness along with the other objectives. The results show that the proposed model and methods allow for the production

*Corresponding author

Email addresses: abm@cs.nott.ac.uk (Ahmad Muklason), ajp@cs.nott.ac.uk (Andrew J. Parkes), exo@cs.nott.ac.uk (Ender Özcan), b.mccollum@qub.ac.uk (Barry McCollum), p.p.mcmullan@qub.ac.uk (Paul McMullan)

of high quality timetable solutions while also providing a trade-off between the standard soft constraints and a desired fairness for each student.

Keywords: Timetabling, Fairness, Multi-objective Optimisation, Metaheuristic
2015 MSC: 00-01, 99-00

---

1. Introduction

Examination timetabling is a well-known and challenging optimisation problem. In addition to requiring feasibility, the quality of an examination timetable is measured by the extent of the soft constraint violations. The formulations for standard examination timetabling problems [1–4] have penalties representing the violations of various soft constraints, including those which influence the spread of examinations across the overall examination time period, providing students with more time for preparation. Of particular interest here is the fact that standard examination timetabling formulations concentrate on minimising the average penalty per student. We believe that this model can lead to unfairness, in that a small but still significant percentage of students may receive much higher than average penalties with a reduced separation between examinations than others. Since students believe that poor timetables could adversely affect academic achievement (as we show later by our survey findings), we believe that overall student satisfaction could be improved by encouraging fairer solutions. In particular, by reducing the number of students that may feel they have been adversely affected for no obvious good reason.

In our prior work [5, 6], we briefly introduced a preliminary extension of the examination timetabling problem formulation in order to encourage fairness among the entire student body (for a study of fairness in course timetabling see [7]). However, the notion of "fairness" in this context is also likely to be quite a complex concept, with no single generic measure appropriate. Hence, to determine student preferences we conducted a survey. This paper reports the main results of the survey and also suggests and analyses extensions to the current models used for optimisation e.g. algorithms are presented along with experimental results.

The contributions of this paper broadly include:

- Presentation of the results of a survey amongst undergraduate and taught-postgraduate students concerning their own preferences for particular properties of examination timetables. These served to confirm our expectation that fairness is indeed a concern for them. In particular, it was apparent that students are mainly concerned with fairness within their immediate cohort.

- An extension to the examination timetabling problem formulation including objectives for fairness. The new problem formulation is inherently multi objective, including both objectives for fairness between all students, and also fairness within specified cohorts.

- Initial work towards building a public repository that extends current benchmark instances with the information needed to build cohorts, thus allowing methods on our formulation to be studied by the community.
- A proposal of an algorithm that works to improve fairness, i.e. a multi-stage approach with weighted Tchebycheff scalarisation technique.
- Initial results on the benchmarks. In particular, we observe that there is the potential to control the trade-off between fairness and other objectives.

The rest of this paper is structured as follows. Section 2 presents the description of the examination timetabling problem and surveys the related works. We then present the findings from the survey, investigating students preferences especially regarding fairness over examination schedules within their immediate cohorts. Section 4 discusses our proposed extension on the examination timetabling problem formulation. The proposed algorithms used within experimentation are introduced in Section 5. Finally the experimental results are discussed in section 6, before the concluding remarks in Section 7.

2. Examination Timetabling

2.1. Problem Formulation

The examination timetabling problem is a subclass of educational timetabling problems. (For example, see the survey of Schaerf [8], where educational timetabling problems are placed within three sub-categories: school timetabling problems, course timetabling problems, and examination timetabling problems). Examination timetabling problems are a combinatorial optimisation problem, in which a set of examinations $E = \{e_1, ..., e_N\}$ are required to be scheduled within a certain number of timeslots or periods $T = \{t_1, ..., t_M\}$ and rooms $R = \{r_1, ..., r_K\}$. The assignments are subject to a variety of hard constraints that must be satisfied and soft constraints that should be minimised [9]. The hard constraints and soft constraints can vary between institutions: examples and detailed explanations can be found in [9].

In order to provide a standard examination timetabling problem formulation as well as the problem datasets from real-world problems in examination timetabling research, some previous studies have shared public benchmark problem datasets. The two most intensively studied benchmark datasets in this research area are the Carter (also known as Toronto) dataset [1] and International Timetabling Competition 2007 (ITC 2007) dataset [10].

The Carter dataset consists of 13 real-world simplified examination timetabling problem instances. The only hard constraint taken into consideration in the Carter model is that whereby each examination has to be allocated a timeslot and be 'clash-free', meaning no student is required to sit more than one examination in the same timeslot. The period (maximum) duration of each timeslot and room capacity are ignored. In other words, it is assumed that each timeslot has a long enough period duration for all examinations and there is always a

room with sufficient capacity to fit all students sitting an examination during each timeslot. A soft constraint violation penalty, called the 'proximity cost', is also introduced. This cost should be minimised in order to give enough period gaps between examinations so as to give students enough time for revision. Formally, the penalty, P, is defined by:

$$P = \frac{\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} C_{ij} W_{|t_j - t_i|}}{Q} \tag{1}$$

where

$$W_{|t_j - t_i|} = \begin{cases} 2^{5-|t_j - t_i|} & \text{iff} 1 \leq |t_j - t_i| \leq 5 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Solutions are subject to the hard constraint which stipulates that no student has two or more exams at the same time:

$$\forall_{i \neq j}. \;\; t_i \neq t_j \;\; \text{when} \;\; C_{ij} > 0 \tag{3}$$

In Equations 1 and 2, given N and Q as the total number of examinations and students respectively, C_{ij} is defined as the number of students taking both examinations i and j, $(i \neq j)$. Also t_i and t_j are the allocated timeslots for examinations i and j respectively, and the timeslots are defined as a time sequence starting from 1 to M, the total number of timeslots.

Furthermore, $W_{|t_j - t_i|}$ is the weight of the penalty produced whenever both examinations i and j are scheduled with $\mid t_j - t_i \mid$ timeslots gap between them. The formula is reasonable in that an increased gap reduces the penalty, but the details are somewhat an ad hoc choice; for example, if the gap between two examinations is greater than five timeslots, then there is no penalty cost.

In contrast with the problem formulation of the Carter dataset, the ITC 2007 dataset formulation allows for the representation of much more complex real-world examination timetabling problems. In addition to the 'clash-free' constraint as required in the Carter dataset, a feasible timetable also requires that each examination has to be allocated to a timeslot with a long enough period duration and at least one room with enough capacity to accommodate all students sitting the examination. One can also specify hard constraints related to period (i.e. examination x has to be timetabled after/same time as/different time to examination y) and hard constraints related to room (i.e. if a room r in a timeslot t is already allocated to examination x, a member of the the specified exclusive examinations, X, then no other examinations can be allocated to room r and timeslot t).

Compared to the Carter dataset, the ITC 2007 examination timetabling formulation has a much richer set of potential soft constraints. Formally, subject to all hard constraints being satisfied, the objective function is to minimise the

total penalty as the result of a weighted sum of soft constraint violations:

$$P = \sum_{s \in S} (w^{2R} C_s^{2R} + w^{2D} C_s^{2D} + w^{PS} C_s^{PS}) + w^{NMD} C^{NMD} + w^{FL} C^{FL} + C^P + C^R \tag{4}$$

Where, the first set is a sum over penalties directly associated to each student s:

C_s^{2R} ('Two in Row') is the penalty incurred whenever a student s has to sit two distinct examinations scheduled in two consecutive timeslots within the same day.

C_s^{2D} ('Two in Day') is the penalty incurred whenever a student s has to sit two distinct examinations scheduled in two non-consecutive timeslots within the same day.

C_s^{PS} ('Period Spread') is the penalty incurred whenever a student s has to sit more than one examination within a specified number of periods.

Other penalties not directly associated to each student are:

C^{NMD} ('Non-Mixed Duration') is the penalty incurred whenever any room in any timeslot is allocated to examinations of differing durations.

C^{FL} ('Front Load') is the penalty incurred by scheduling what are considered large examinations towards the end of the examination period.

C^P is the penalty associated to a period/timeslot whenever it is used for examinations.

C^R is the penalty associated to a given room whenever it is allocated to examinations.

The weighting applied to each of the individual penalties listed, e.g. w^{2R}, as well as the other specifications, e.g. the penalty associated to each room/timeslot, are defined in the 'institutional model index' file. Full details, including the mathematical programming formulation of this problem are found in [10].

The other examination timetabling problem instances reported in the literature include benchmark datasets generated from University of Nottingham [11], University of Melbourne [12], MARA University Malaysia [13], Universiti Kebangsaan Malaysia (UKM) [14], University of Yeditepe [15], Universiti Malaysia Pahang [16], and KAHO Sint-Lieven [17].

2.2. Related Work

Examination timetabling problems have attracted researchers over the last number of decades, in particular those within the area of operation research and artificial intelligence. The real-world problems can become even more challenging and complicated due to the increasing tendency of many universities to offer

cross-disciplinary programs, although there have been many successfully implemented approaches reported in the literature in solving these problems. These approaches range from traditional graph colouring heuristics, to meta-heuristics and hyper-heuristics.

Surveys on the state-of-the-art examination timetabling problem formulations, techniques and algorithms have been reported in prior work such as [8, 18–20]. In [20], which could be considered the most comprehensive survey, the existing approaches/techniques are classified into the following categories; clustering methods, constraint-based methods, meta-heuristics, multi-criteria techniques and hyper-heuristics.

A hyper-heuristic is a high level search method that selects or generates problem specific low-level heuristics for solving computationally difficult combinatorial optimisation problems [21]. A key potential benefit of hyper-heuristics is that they have reusable components and can handle a variety of problem instances with different characteristics without requiring expert intervention. See [21] for a recent survey on hyper-heuristics. Here, we provide an overview of selection hyper-heuristics for solving examination timetabling problems.

Currently, selection hyper-heuristics generally use a single point-based search framework. They process a single solution at a time, remembering the best found solution so far. An initially generated solution is fed through an iterative cycle until a termination criterion is satisfied, in an attempt to improve the solution quality with respect to a given objective. There are two main methods employed at each step, each playing a crucial role in the success of the overall performance of a selection hyper-heuristic. Firstly, a heuristic selection method is employed to choose a low level heuristic. After the application of the selected heuristic to the current solution, a new solution is obtained. Secondly, the move acceptance strategy decides whether to accept or reject that new solution. Of course, such a structure is also present in many meta-heuristics. However, the point of a hyper-heuristic is to provide a modular architecture and enable such structures to be explicitly separated from the details of individual problem domains; hence, aiming to make it easier to exploit advanced intelligent adaptive methods (e.g. see [22, 23]).

Although the study and application of hyper-heuristics is a relatively new research area, they have been successfully applied to solve many combinatorial optimisation problems. One of the most successful implementations of hyper-heuristics is in timetabling problems, in particular examination timetabling. Most recently published studies on examination timetabling problems with hyper-heuristics are discussed in [17, 20, 24–33].

Bilgin et al. in [24] carried out an empirical analysis of the performance of hyper-heuristics with differing combinations of low-level heuristic selection and move acceptance strategies over examination timetabling problem benchmark instances. The heuristic selection strategies consist of seven methods; simple random, random descent, random permutation, random permutation descent, choice function, tabu search, and greedy search. The move acceptance strategies comprise five methods; all moves accepted (AM), only improving moves accepted (OI), improving and equal moves accepted (IE), great deluge and Monte Carlo

strategy. This combination of low-level heuristics selection and move acceptance strategies result in 35 different possible hyper-heuristics. To evaluate the performance of hyper-heuristics, the study was carried out over 14 well-known benchmark functions as well as 21 examination timetabling problem instances from the Carter benchmark dataset [1] and Yeditepe benchmark dataset [15]. The experimental results showed that the combination of choice function as a low-level heuristic selection strategy and monte carlo [34] as move acceptance strategy is superior to the other combinations.

Graph-based hyper-heuristics incorporating tabu search (TS) evaluated over the Carter dataset reported good results in [25]. Further, in [35] the graph-based hyper-heuristics incorporated with steepest descent method (SDM), iterated local search (ILS), and variable neighbourhood search (VNS) were also implemented with the Carter dataset. The computational results showed that iterative technique e.g. VNS and ILS were more effective than TS and SDM.

In addition, hyper-heuristics with late acceptance strategy were studied in [26]. Within this strategy, in order to decide whether to accept a new candidate solution, it is compared with solutions from earlier iterations rather than with the current best solution. The proposed approach was tested over the Carter dataset. The experimental study showed that the late acceptance strategy is best suited with simple random low-level heuristic selections. This combination outperforms the combination of late acceptance strategy with reinforcement learning or statistical based heuristic selection.

An evolutionary algorithm based hyper-heuristic for examination timetabling problem with the Carter dataset was studied in [27]. The study examined three different proposed representations of low-level heuristics combinations; fixed length heuristic combination (FHC), variable length heuristic combination (VHC), and N-times heuristic combination (NHC). The experimental results showed that NHC and VHC perform much better than FHC. The results also showed that the combination of the three representations yields better performance than FHC, VHC, and NHC alone.

Burke et al. [28] compared the performance of different Monte Carlo based hyper-heuristics over the Carter dataset. Four low-level heuristic selection methods were evaluated; simple random, greedy, choice function and learning scheme, and three Monte Carlo based move acceptance methods; standard simulated annealing, simulated annealing with reheating, and exponential Monte Carlo. The results indicated the success of a hyper-heuristic combining a reinforcement learning based method, namely choice function and simulated annealing with reheating.

Tournament-based hyper-heuristics for examination timetabling problems were investigated in [17]. The study evaluated tournament based random selection of low-level heuristics coupled with four move acceptance criteria; 'improving or equal', simulated annealing, great deluge, and an adapted version of the late acceptance strategy. The proposed hyper-heuristics were tested over three benchmark datasets, namely Carter, ITC 2007, and KAHO datasets. The KAHO dataset is a new examination timetabling problem benchmark, unique to prior problem instances, in that there are two types of examinations, i.e. written

and oral examinations. Tested over the Carter dataset, the experimental results showed that the proposed approach could improve the best known solutions in the literature, i.e. 7 out 13 problem instances. However, over the ITC dataset, it failed to improve on the results of best known approaches in the literature, but nonetheless could still produce competitive results.

In [30], in order to assign exams to time slots and rooms, bin packing heuristics were hybridised under a random iterative hyper-heuristic. The experiments over the ITC 2007 dataset showed that combining the heuristics, which perform well when they are utilised individually, could produce the best solution. The proposed approach was reported to produce solutions competitive with the best known approaches reported in the literature.

Abdul-Rahman et al. in [31] introduced an adaptive decomposition and heuristic ordering approach. In the process of assignment, the examinations are divided into two subsets, namely difficult and easy examinations. Moreover, in order to determine which examination should be assigned to a timeslot first, the examinations are ordered based on differing strategies of graph colouring heuristics. Initially, all examinations form the set of easy examinations. Then, during the process of assignment, if an examination could not be assigned to any feasible timeslot, it is moved to the subset of hard examinations. This process is repeated until all examinations are assigned to feasible timeslots. The experimental study on the Carter dataset showed that the proposed approach is competitive with other approaches.

In [32] a constructive approach, termed linear combination of heuristics and based on squeaky wheel optimisation [36] was proposed. During the assignment process, each examination is associated with a difficulty score based on a graph colouring heuristic and a heuristic modifier which changes dynamically in time. The examinations are ordered by their associated difficulty score. The examination with the highest difficulty score will be assigned resources (i.e. timeslot and room) before other lower scoring (less difficult) examinations. Initially, the difficulty score of an examination is set to be equal to its order by the chosen graph-colouring heuristic, then its difficulty score is increased using the heuristics modifier function whenever a feasible resource assignment is not possible. The cyclic process stops whenever a feasible solution is obtained. In order to get a high quality feasible solution, a resource is allocated from those incurring the least penalty. Testing over the Carter and ITC 2007 datasets showed that in addition to its simplicity and practicality, the proposed approach delivers a comparable performance to the previously reported approaches.

In [33], a hyper-heuristic with a heuristic selection mechanism using a dynamic multi-armed-bandit extreme value-based reward scheme was proposed. The move acceptance criteria are generated automatically using the proposed gene expression programming framework. The proposed approach was tested on two different problem domains, namely the ITC 2007 examination timetabling problem and dynamic vehicle routing. The experimental results showed that the proposed approach outperforms the ITC 2007 winner as well as post-ITC 2007 methods on 4 out of 8 problem instances.

2.3. Fairness in Timetabling

The concept of fairness (also 'balance' or 'evenness') has been extensively investigated in the field of political science and political economics. Some common sense definitions of fairness in these fields are discussed in [37] and [38]. In [37], fairness is defined as an allocation where no person in the economy prefers anyone else's consumption bundle over his own, whilst [38] defines fairness as a fair allocation that is free of envy. The fairness issues have been well studied in the field of computer networks in areas such as fair resource distribution among entities [39–43] and fair congestion control [44, 45]. In the field of operation research, fairness issues have been investigated for particular problems, for example, in flight landing scheduling (see [46] and [47]).

However, there is a limited number of prior studies explicitly dealing with fairness issues in timetabling. Aiman et al. [48] discussed the results from a survey conducted among nurses in Malaysian public hospitals, emphasizing the importance of fairness in rosters to the nurses in terms of workload balance and respecting their preferences. Smet et al. [49] proposed the use of a fairness model within objective functions to produce fair nurse rosters and tested a hyper-heuristic approach [50] for solving a nurse rostering problem for Belgian hospitals. The results indicated that fairness can be achieved at the expense of a slightly higher overall objective value measured with respect to the generic objective function.

Martin et al. [51] tested a range of fairness models embedded into objective functions under a cooperative search framework combining different (hyper/meta-)heuristics for fair nurse rostering using the Belgian hospital benchmark [50]. From the results, it was shown that each cooperating metaheuristic using a different fairness model yields the fairest rosters under the proposed distributed framework.

Castro and Manzano [52] proposed a formulation of the balanced academic curriculum problem which requires assignment of courses (modules) to periods for teaching while respecting the prerequisite structure among the courses and balancing the student's load - which can be considered as a fairness issue. This formulation is later extended by Gaspero and Schaerf [53] and Chiarandini et al. [54].

The most relevant work was presented on fairness in course timetabling by [7, 55]. The authors proposed a simulated annealing algorithm variant using single and bi-objective course timetabling formulations based on max-min fairness [56] and Jain's fairness index [57] respectively. The experimental results on a set of curriculum-based course timetabling instances, including the ITC2007 benchmark [58], showed that fairer solutions can be produced in exchange for a relatively small increase in the overall number of soft constraint violations.

To the best of our knowledge this study, combined with our earlier initial studies and brief reports [5, 6], is the first extensive study of fairness in examination timetabling.

3. Students Perspective on Fairness: A Survey

Some surveys focussing on preferences within examination timetabling have previously been conducted, the first of particular interest involving University registrars [9]. A later survey [59] was directed at students and invigilators; as might be expected, it was found that "students felt that the most important consideration while preparing the timetable is to have a uniform distribution of exams over the examination period". However, as indicated earlier, under the current construction methods, a percentage of students will almost certainly have poorer distributions than others. The previous surveys had not covered all aspects of student preferences on how such potential unfairness should be managed. Hence, we conducted a survey to give a deeper understanding of their preferences on the fairness and nature of the distribution of exams.

In the survey reported on here, in addition to general questions regarding students personal experience, the questionnaire consisted of two main parts. The first part was concerned with students perspective on the fairness issue in relation to the general examination process, while the second part was concerned with students detailed personal preferences on their own exam timetable.

In the first part, students were surveyed on their opinion regarding fairness in general as well as how they understood and defined fairness in relation to their examination timetable. The survey included questions on whether fairness should only be enforced among the entire student body within the university or also among students within the same course. (In this paper, we use the terminology that a 'course' is a set of different 'modules', spread over many terms or semesters, and forming a degree - also called a 'programme').

In the second part of the survey, the students were asked about their detailed preferences on how their examinations are timetabled. These included preferences regarding the time of the examinations and the gap between them. Moreover, the questionnaire also asked students to consider the "difficulty" (with regard to amount of preparation/revision required) of their exams. To the best of our knowledge, the difficulty of an exam was neglected in the state-of-the-art examination timetabling formulations. In the prior problem formulations, all exams were assumed to have the same level of difficulty. Also included was an investigation into how students would penalise the gap between two of their exams, in comparison with the equivalent Carter problem formulation. In this, the gap between two exams are penalised 2^{5-gap} when the gap is 1-5 timeslots (see Equation 1). Overall, the survey aimed at getting some insight into student preferences in order to construct a more representative examination timetabling problem model for real-world cases.

3.1. Survey Results

The feedback data had been collected from 50 undergraduate and taught postgraduate students at The University of Nottingham in April 2014 regarding their autumn term 2013/2014 examinations. From the questionnaire feedback, the most significant findings are as follows.

From the general response, it was found that the average number of examinations students had during the examination session was four examinations within 10 days. With respect to their examination timetables, it was found that only 40% of the students were happy or very happy and 14% of them were unhappy. Even, 28% of respondents believed that their examination timetable negatively affected their academic achievement. The common reasons that made them unhappy were; examinations timings that are too close to each other (less than 24 hours gap between exams) especially if one or both of the exams are difficult; locations that are different from the base campus; and having an exam in the last day of the examination period.

In response to the fairness issue, our survey revealed that 10% of students think that the examination timetable is unfair amongst students, 60% of students think it is fair, with the rest neutral. However, as expected, almost all students agreed that the examination timetable should in principle be fair. Regarding the scope of fairness, as summarised in Table 1, it is shown that 36% of respondents strongly agreed and 46% agreed that examination timetables should be fair amongst students taking the same exams. Furthermore, when the respondents were asked to detail their perception with respect to the scope of fairness, 42% strongly agreed and 42% agreed that examination timetables should be fair amongst students enrolled on the same course. Interestingly, the statistic changed with 24% strongly agreed and 42% agreed, if they were asked whether the examination timetable should be fair amongst the entire student body of the university (though enrolled on different courses). This finding indicates that fairness within a course is more crucial than fairness amongst the entire student body of the university. This is considered as a natural response as students on the same course are colleagues but are also competing against each other. Dissatisfaction may therefore arise when a student knows that a fellow student has much more time for revision before an important or difficult exam.

Table 1: Students response regarding fairness: whether fairness should be enforced in different scenarios (% of students). Note: DS=Disagree Strongly, D=Disagree, N=Neutral, A=Agree, AS=Agree Strongly.

Fairness among students:	Students Response (%)				
	Disagree Strongly DS	D	N	A	AS
Taking the same exam	2	2	14	46	36
Taking the same course	2	4	10	42	42
Overall, though different course	2	8	24	42	24

Note that the notion of 'within a course' may be extended to 'within a cohort' with various different choices for cohorts. For example, a 'cohort' could refer to 'year of study', and justified on the grounds that fairness between final year students is more important than for first years (as the exams typically contribute more to the final degree).

Further findings in our survey relate to what students think about the quality of timetables, in which students personal preference over their examination

timetable are investigated. We found some factors that affected students preferences. Overall, it is not surprising that almost 3 in 4 students (74%) preferred to have exams that are spread out evenly throughout the examination period as opposed to only 12% who preferred exams to be 'bunched' together within the period. When students were asked to make a trade-off between the total examination period length and the gap between exams, in which a shorter total exam period would mean a reduced gap between exams, 40% preferred a longer total examination period while 12% preferred the opposite with the rest preferring no change. Furthermore, 82% were not willing to accept more than one exam in a day even if this was the same (fair) for all students. It confirms that it is seen as critical for students to have sufficient gaps between their exams and as more important than overall fairness.

In relation to exams and the allocation of timeslots, assuming that there are three timeslots a day, the afternoon (middle) session was the most preferred, while morning and evening session ranked second and third respectively. In addition, some students (31%) preferred to have no exam on particular days of the week. The least preferred days were Saturday or Sunday, Friday, Monday, and any day after a student has attended an exam. More than half (54%) preferred to have no exam on the weekend.

In the current state-of-the-art exam timetabling problem formulation, the exams are assumed as having equal difficulty levels. In contrast, our findings showed that 53% of the students strongly agreed and 37% of them agreed that some examinations are more difficult than the others. Thus, these exams should be scheduled with longer gaps for students to allow for preparation. Furthermore, 50% of students preferred difficult exams to be scheduled earlier while only 20% preferred the opposite.

In order to determine what students consider the ideal length for gaps between exams, the students were asked to provide a penalty value (0-9) according to a set of possible exam schedule options, as follows. Given two exams, three days exam period with three timeslots per day (morning, afternoon, and evening), if the first exam was scheduled in the first timeslot i.e. morning of the first day, students were asked to indicate a penalty expressing their unwillingness if the second exam was scheduled in the second timeslot of the first day, the third timeslot of the first day, and so forth until the third timeslot of the third day.

For each schedule option, the two exams are set up in three different scenarios. In the first scenario both exams are assumed to have the same difficulty level, while in the second scenario the first exam is assumed as an easy exam and the second exam difficult. Contrasting with the second scenario, in the third scenario the first exam is assumed as the difficult exam and the second easy. The average penalty given by the respondents over these three scenarios is summarised by Figure 1.

The x-axis in Figure 1 indicates each option for the scheduling of the second exam, given that the first exam is scheduled in the first timeslot of the first day, while the y-axis indicates the penalty. In x-axis, $D1.T2$ represents the first day, second timeslot, $D2.T1$ represents the second day, first timeslot and so on. The

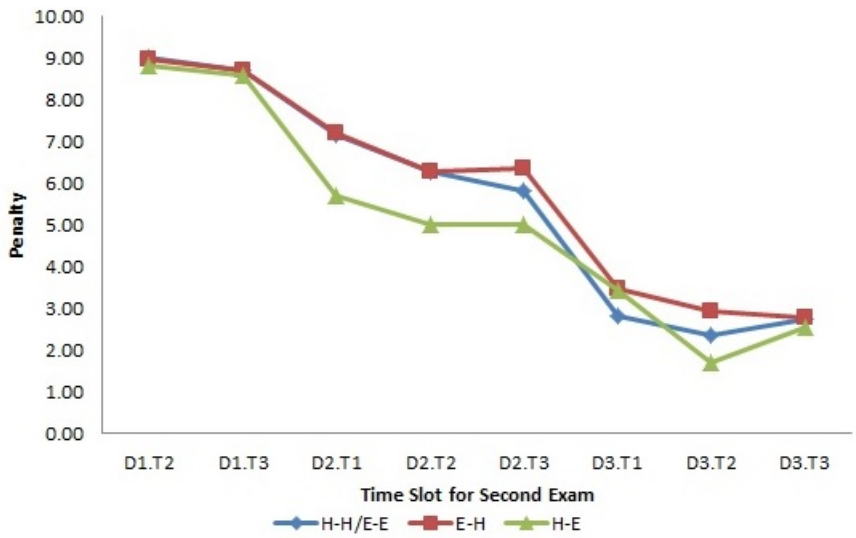

Figure 1: Penalty Given By Students. Given two exams with different level of difficulty: easy (E) and hard (H); and three scenarios

score 0 in y-axis means that students have no problem with the timetable while 9 means students really don't expect that timetable.

From Figure 1 we know that no student expects to have two exams in the same day. We also observe that between the easy and difficult exams, the students expect more gaps than with the reverse. This is understandable given that students need more time for preparation for a more difficult exam.

An additional challenge with accounting for this is the need to determine perceptions of the difficulty of examinations. This measure may be determined by obtaining the students opinion after taking the examinations, or asking samples of students in advance to nominate which examinations needed more preparation time.

4. Towards an Extended Formulation of Examination Timetabling with Fairness

A commonly used fairness measure is the 'Jain's Fairness Index' (JFI) [57]. Suppose a set A of students, has associated penalties $P(A) = \{p_i\}$, with mean value, $\bar{P}$, and variance σ_P^2. Then a reasonable measure of the width, and so fairness, is the standard 'Relative Standard Deviation' (RSD) defined by $RSD^2 = \sigma_P^2/\bar{P}^2$. The JFI over all students in A, which throughout this paper is referred to as $JFI(A)$, is then a convenient non-linear function of the RSD:

$$JFI(A) = \left(1 + RSD^2\right)^{-1} = \frac{\left(\sum_{i \in A} p_i\right)^2}{|A| \sum_{i \in A} p_i^2} \tag{5}$$

and it is (arguably) 'intuitive' as it lies in the range $(0, 1]$ assuming that there is at least one non-zero p_i and a totally fair solution (all penalties equal) has JFI=1. A solution with no penalties is treated separately and assumed to have JFI=1 as well.

Moreover, for a course/cohort, C_k, the 'fairness within a course/cohort', which throughout this paper is referred $JFI(C_k)$, can be defined by simply limiting to the penalties for the students within C_k rather than all students in the university. A candidate objective function to enhance fairness within cohorts is then simply the sum or average of JFI values per cohort:

$$\text{(maximise)} \quad \sum_k JFI(C_k) \tag{6}$$

As an illustration, consider the case of two cohorts with two (groups of) students each, and with $P1$ and $P2$ as the set of penalties for cohorts 1 and 2 respectively. Suppose there are two candidate solutions $S1$ and $S2$ with values:

Soln	**P1**	**P2**	**avg(P)**	**JFI(A)**	**J1**	**J2**	**JFI(C)**
S1	{4,4}	{2,2}	**3**	0.9	1.0	1.0	**1.0**
S2	{4,2}	{4,2}	**3**	0.9	0.9	0.9	0.9

where $JFI(A)$ is the JFI over all the students, $J1$ and $J2$ are the JFI values for cohort 1 and cohort 2 respectively, and $JFI(C)$ is the average JFI within a cohort. The two solutions have the same overall average penalty, $avg(P)$, and overall fairness, $JFI(A)$. However, we believe that students would prefer solution $S1$ as it is fairer within each cohort, and this is captured by the higher value of $JFI(C)$. Of course, the situation will not always be so simple. Consider a second example but with three students per cohort, and three solutions as follows:

Soln	**P1**	**P2**	**avg(P)**	**JFI(A)**	**J1**	**J2**	**JFI(C)**
S1	{8,8,9}	{2,2,2}	5.2	0.725	0.997	1.0	**0.998**
S2	{8,8,2}	{8,2,2}	**5.0**	0.735	0.818	0.667	0.742
S3	{7,7,9}	{4,3,3}	5.5	**0.852**	0.985	0.980	0.983

$S2$ is the lowest overall penalty and would be the standard choice, but is not the fairest both overall and within the cohorts. Potentially, $S1$ might be preferred because it is the most fair within the cohorts, or alternatively $S3$ as it is most fair between all the students. This suggests there should be a trade-off between overall total penalty, overall fairness, and fairness within cohorts. Note that alternatives to the objective function in (7) should also be considered; e.g. for some suitable value of p, to simply minimise the sum of p'th powers of RSDs:

$$\text{(minimise)} \quad \sum_k RSD^p(C_k) \tag{7}$$

or maybe even use an extended version of the JFI with $JFI_p = (1 + RSD^p)^{-1}$.

Lastly, for the 'hardness', of exams, we propose to simply give a difficulty index for each exam and use this in modified definitions of penalties, e.g. having an exam scheduled the day before a difficult exam is penalised harder than if it were scheduled before an easy exam. The difficulty index is formulated in Equation 11, 12, and 13.

Adapted from [10], suppose E is a set of exams, S is a set of students and P is total number of periods. We use three binary variables y_{pq}, t_{is} and X_{ip}^{P} defined by:

$$y_{pq} = \begin{cases} 1 & \text{iff periods } p \text{ and } q \text{ are on the same day} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

$$t_{is} = \begin{cases} 1 & \text{iff student } s \text{ is enrolled in exam } i \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

$$X_{ip}^{P} = \begin{cases} 1 & \text{iff exam } i \text{ is scheduled in period } p \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

Given extra data in the from of difficulty indices d_i for each exam i with values ranging between 1 and 3 expressing exam difficulty (e.g. 1 = easy, 2 = medium, 3 = hard), the modified 'two exams in a row', 'two exams in a day', and 'period spread' penalties are defined here.

Two Exams in a Row Penalty

Provided that student s enrolled in both exams i and j (two distinct exams), and exam j is scheduled on the same day and immediately after exam i, two exams in a row (C_s^{TR}) penalty is defined as follows:

$$C_s^{TR} = \sum_{\substack{i,j \in E \\ i \neq j}} \sum_{\substack{p,q \in P \\ q=p+1 \ \& \ y_{pq}=1}} W^{TR}(d_i, d_j) t_{is} t_{js} X_{ip}^{P} X_{jq}^{P} \tag{11}$$

where $W^{TR}(d_i, d_j)$ is a matrix of penalty values. Note that it is not necessarily symmetric, e.g. to allow different preferences for 'easy then difficult' and 'difficult then easy' in the exam sequence.

Two Exams in a Day Penalty

Similar conditions to that of two exams in a row (C_s^{TR}) penalty apart from the fact that exam j and i are not scheduled in two consecutive periods. C_s^{TD} is defined as:

$$C_s^{TD} = \sum_{\substack{i,j \in E \\ i \neq j}} \sum_{\substack{p,q \in P \\ q>p+1 \ \& \ y_{pq}=1}} W^{TD}(d_i, d_j) t_{is} t_{js} X_{ip}^{P} X_{jq}^{P} \tag{12}$$

Period Spread Penalty

Given that student s enrolled in both exams i and j (two distinct exams) and g is the period gap between i and j, C_s^{PS} is defined as:

$$C_s^{PS} = \sum_{\substack{i,j \in E \\ i \neq j}} \sum_{\substack{p,q \in P \\ p < q \leq p+g}} W^{PS}(d_i, d_j) t_{is} t_{js} X_{ip}^P X_{jq}^P \tag{13}$$

with associated matrices $W^{TD}(d_i, d_j)$ and $W^{PS}(d_i, d_j)$ of penalty parameters.

Unfortunately, since we don't have data on examination difficulty, we have not directly studied this particular extension to the examination timetabling formulation. However, we expect that it would be straightforward to extend standard algorithms to cope with these 'examination difficulty' aspects. Therefore we would encourage the collection of such data whenever possible, e.g. by providing students a form with which to weight the difficulty of each exam.

5. A Multi-phase Approach for Fairer Examination Timetables

Our main intent in this paper is to study the potential for a trade-off between the standard objectives and those dealing with fairness. Hence, we need to be able to find good solutions, not giving up too much on the standard objectives, but also incorporating fairness. Accordingly, the proposed approach used here for solving the examination timetabling problems with fairness consists of three consecutive phases. Phase 1 aims at producing an initial feasible solution, i.e satisfying all hard constraints, while phase 2 aims at improving the quality of the initial solution in terms of standard objective function. Finally, phase 3 attempts to make the solutions fairer, whilst staying in a 'reasonable region' of the Pareto Front.

In phase 1, initial feasible solutions are constructed using an adaptive heuristic ordering approach. We adapted squeaky wheel optimisation [36] involving heuristic ordering as proposed in [32, 60]. In phase 2 and phase 3, a selection hyper-heuristic is employed, embedding *reinforcement learning* and *great deluge algorithm* as heuristic selection and move acceptance components respectively. This method is adapted from [61] with some modifications. We employed 14 low-level heuristics commonly used in the literature for examination timetabling problems as provided in Table 2. The low-level heuristics in Table 2 are quite obvious except for the Kempe-chain move [62], which involves moves within two full sets of exams and in a way that guarantees conflict freedom.

Table 2: Perturbation Low-Level Heuristics (LLHs) for Exam Timetabling Problems

LLH	**Description**
LLH_1	Select one exam at random and move to a new random feasible timeslot and a new random feasible room.
LLH_2	Select two exams at random and move each exam to a new random feasible timeslot.
LLH_3	Select three exams at random and move each exam to a new random feasible timeslot.
LLH_4	Select four exams at random and move each exam to a new random feasible timeslot.
LLH_5	Select two exams at random and swap the timeslots between these two exams while maintaining the feasibility of the two exams.
LLH_6	Select one exam at random and select another timeslot then apply the Kempe-chain move.
LLH_7	Select one highest penalty exam selected from a random 10% selection of the exams and select another timeslot then apply the Kempe-chain move.
LLH_8	Select one highest penalty exam selected from a random 20% selection of the exams and select another timeslot then apply the Kempe-chain move.
LLH_9	Select two timeslots at random and swap the exams between them.
LLH_{10}	Select one timeslot at random and move the exams assigned to that timeslot to a new feasible time-slot.
LLH_{11}	Shuffle all time-slots at random.
LLH_{12}	Select one exam at random and move it to a randomly selected feasible room.
LLH_{13}	Select two exams at random and swap their rooms (if feasible).
LLH_{14}	Select one large exam at random and move to a new random earlier feasible timeslot.

The selection hyper-heuristic method and problem domain components, including all low level heuristics are implemented as a part of a hyper-heuristic framework referred to as HyFlex [63, 64] which is designed for rapid development and evaluation of hyper-heuristics. The ITC 2007 problem specification is used as a basis to implement the components of the examination timetabling problem domain. For example, the objective function is the standard objective function (disregarding fairness) as specified in Equation 2 for the ITC 2007 dataset and Equation 4 for the Carter dataset.

The reinforcement learning heuristic selection simply gives each low level heuristic a reward or punishment. Initially, each low-level heuristic receives the same score (e.g. 10, in this case). After the application of a chosen low level heuristic, if the objective function value remains the same or has improved, the score of the relevant heuristic is increased by 1 until an upper bound is reached (e.g. 20, in this case). Similarly, if the solution has become worse, the score of the relevant heuristic is decreased by 1 until the lower bound score (e.g. 0, in this

case) is reached. In each iteration, a low-level heuristic with the highest score is chosen. If there is a tie between low-level heuristic scores, then one of them is selected randomly. See [61] for a study on how different parameter settings (e.g. reward and mechanism procedure, lower and upper bound) influence the overall performance of an algorithm.

The great deluge method is a threshold move acceptance method. This method accepts a new solution obtained after the application of a chosen low-level heuristic, if it is no worse than the current solution or given threshold level. Initially, the threshold level is set to the objective function value of the initial solution. Then, at each iteration, the threshold level is decreased gradually by the *decay rate*. In our experiment, the decay rate is initially set to 0.001; a value experimentally known to be reasonable. Generally, the decay rate could be set as the difference between threshold level and the desired objective function value divided by the number of iterations, as in [65]

A feasible solution is constructed during phase 1 which is then fed into phase 2. Although phase 2 and 3 use the same selection hyper-heuristic method, they are structured to improve the quality of a solution in terms of different objectives. Phase 2 uses the standard penalty as the objective while phase 3 considers both the standard penalty and fairness. The simplest approach within phase 3 is by treating fairness i.e. $JFI(A)$ as an objective function and adding 'not worsening the standard penalty' as a hard constraint. However, as shown by our prior work [5, 6], it might be impossible in practice to improve fairness without worsening the standard penalty; we need to capture the best trade-off between standard penalty and fairness.

Algorithm 1 Pseudo-code for Improving Initial Feasible Solution

```
procedure IMPROVESOL(Initial solution I, Time limit T, Set of low-level heuristic (llh) H)
    Initialise Pareto Solution set P ← ∅
    //set current solution
    C ← I
    //set best solution
    C_b ← I
    //set boundary level
    B ← getFunctionValue(C)
    //set decay rate
    α ← 0.001
    //set score for each low-level heuristic (llh) equal to 10
    Set an array of integer,G ← new int[H.size]
    for j=0,j=H.size do
        G[j]← 10
    end for
    while not exceed T do
        //get the index of low-level heuristic with highest score
        l ← getBestLLH(H)
        //apply l over C to generate new solution C*
        C* ← applyHeur(l,C)
        v ← getFunctionValue(C)
        v* ← getFunctionValue(C*)
        if v* ≤ v OR v* ≤ B then
            //accept the new solution
            C ← C*
            if v* < getFunctionValue(C_b) then
                C_b ← C*
            end if
            if G[l] < 20 then
                G[l] ← G[l]+1
            end if
        else
            if G[l] > 0 then
                G[l] ← G[l]-1
            end if
        end if
        B ← B-α
    end while
    //return the best solution
    return C_b
end procedure
```

5.2. Phase 3: Enforcing Fairness

In our prior work [5, 6], a modified objective function was proposed in order to enforce fairness within the obtained solutions. Instead of 'linear summation' of soft constraint violations associated with each student, 'summation of power' was introduced. Experimental results on the Carter dataset showed that the approach can produce fairer solutions with a small increase in the average penalty.

The limitations of 'summation of power' approach is that in each single run, it only produces a single solution. In addition, it also requires significantly (approximately 28 times) higher computational times compared to the original linear summation objective function. Therefore, to cope with these limitations, in this paper we study a different approach, namely, a multi-criteria / multi-objective optimisation approach on a large set of examination timetabling problem instances with various characteristics from three well-known benchmarks. Within the proposed approach, the standard penalty is minimised, while $JFI(A)$ is maximised. As discussed in the previous section, the standard penalty is defined in equation 2 for the Carter dataset and equation 4 for the ITC 2007 and Yeditepe datasets, while $JFI(A)$ is defined in equation 5. For simplicity of illustration, the second objective is also turned into a minimising function and reformulated in Equation 14 as an unfairness measure, $AJFI$.

$$AJFI(A) = 1 - JFI(A) \tag{14}$$

Since we consider the problem as a multi-objective instead of single-objective problem, the output of the algorithm in this phase is a set of approximation Pareto optimal solutions instead of a single solution. The algorithm used to generate approximation Pareto optimal solutions in this study is presented in Algorithm 2 below. Basically, the algorithm is a hybridisation of reinforcement learning and the great deluge algorithm. To cope with the multi-objective nature of the problem, a classical scalarisation method, namely weighted Tchebycheff [66] is employed as a new objective function. This function requires an initial set up weight and reference point, which dictates the ideal objective function value to be achieved, for each objective function.

Suppose f_1 and f_2 are the two objectives with their respective weights i.e. w_1 and w_2 ($w_1 + w_2 = 1$) and respective reference points i.e. r_1 and r_2. The weighted Tchebycheff function is given in Equation 15. This equation could be generalised to any number of objective functions.

$$\text{minimise}[\max(|f_1(x) - r_1(x)|w_1, |f_2(x) - r_2(x)|w_2)] \tag{15}$$

As shown by Algorithm 2, the algorithm consists of outer iterations (line 3) and inner iterations (line 23). In each outer iteration, the weight vector is generated randomly while the current solution (line 9) is set to a random solution from aggregate Pareto set($\mathtt{P}_a$). We have conducted preliminary experiments comparing the setting of the current solution to the initial solution, best solution found so far (in term of weighted Tchebycheff value), and a random solution

from the aggregate Pareto set. The experimental results showed that setting the current solution to a random solution from the aggregate Pareto set results in the best approximation Pareto optimal solutions.

Furthermore, the reference points in this algorithm are set to `80%` of the objective function values of the initial solution (see line `14-15`). This value is chosen from our preliminary experiments with 80% results in the best approximation of Pareto optimal solutions compared to 60%, 70%, and 90%.

For each inner iteration within a single outer iteration, each move (applying a low-level heuristic) results in a new solution. In this stage, we have two alternatives, i.e. adding any new solutions to the Pareto set ($\mathtt{P}_i$) or only adding accepted solutions, which are improving the current solution or better than boundary level, to the Pareto set. Our preliminary experiment showed that adding any new solutions to the Pareto set results in a better approximation Pareto optimal solution set. After the last inner iteration, the Pareto set is sorted using the Kung Pareto sorting algorithm [67][1] to generate a sorted Pareto optimal solution set (see line `44` in Algorithm 2), which is the set of non-dominated solutions. The sorted Pareto solutions from a single outer iteration ($\mathtt{P}_i^*$ are then added to the aggregate Pareto optimal solution ($\mathtt{P}_a$). Finally, the sorted aggregate Pareto solutions ($\mathtt{P}_a^*$) form the final approximation Pareto optimal solutions.

By employing multiple outer iterations, this can produce more Pareto solutions by aggregating the Pareto solution set. It is useful to note that a single outer iteration of the algorithm in itself could produce a set of Pareto solutions as opposed to a single solution.

Since each objective function has different value ranges, the aspect of normalisation is worth noting. Our preliminary experiment showed that normalising the objective function values to `0-1`, with the initial current solution and reference point as lower bound respectively, could improve the quality of approximate Pareto optimal solutions.

6. Experiments and Discussion

6.1. Experimental Data and Settings

The experiments were conducted over three different real-world examination timetabling benchmark problem datasets, namely Carter [1], ITC 2007 [58] and Yeditepe [2, 15]. The properties of these datasets are summarised in Table 3.

In our experiment, the original format of the Carter and Yeditepe datasets were converted into ITC 2007 format, so that the same solver could be applied to all problem instances. Moreover, the data format was extended to provide more information to support handling fairness, e.g. information about students

[1]Note, although decades old, this algorithm is still considered an efficient and widely used Pareto sorting algorithm, i.e. $O(NlogN)$ for $k = 2$ and $k = 3$ and complexity $O(Nlog^{k2}XN)$ for $k > 3$, in which k is the number of objectives. In any case, this Pareto Sorting is only a small component of our proposed algorithm, and so improved methods would not impact on the results as the size of Pareto set is not very large.

Algorithm 2 Pseudo-code for Generating Pareto 'Optimal' Solutions

```
procedure generateParetoOptSol(Number of iterations N, Initial solution I, Time limit per iteration T, Set of low-level heuristics (llh) H)
    Initialise aggregate Pareto solution set, P_a ←I
    for i← 1, N do
        Initialise Pareto Solution set P_i ← I
        //Generate weight with real random number [0-1]
        w_1 ← genRandNum(0,1)
        w_2 ← 1-w_1
        //set current solution
        C ← I
        //set the reference points for the first and second objective function
        //f_1 and f_2 are the first and second objective function values
        r_1 ←0.8 f_1(C)
        r_2 ←0.8 f_2(C)
        //set boundary level
        B ← getTchebycheffSum(f_1(C),f_2(C),r_1,r_2,w_1,w_2)
        //set decay rate
        α ← 0.001
        //set score for each low-level heuristic (llh) equal to 10
        Set an array of integer,G ← new int[H.size]
        for j=0,j=H.size do
            G[j]← 10
        end for
        while not exceed T do
            //get the index of low-level heuristic (LLH) with highest score
            l ← getBestLLH(H)
            //apply LLH with index l over C to generate new solution C*
            C* ← applyHeur(l,C)
            v ← getTchebycheffSum(f_1(C),f_2(C),r_1,r_2,w_1,w_2)
            v* ← getTchebycheffSum(f_1(C*),f_2(C*),r_1,r_2,w_1,w_2)
            if v* ≤ v OR v* ≤ B then
                //accept the new solution
                C ← C*
                if G[l] < 20 then
                    G[l] ← G[l]+1
                end if
                P_i ← P_i∪ C*
            else
                if G[l] > 0 then
                    G[l] ← G[l]-1
                end if
            end if
            B ← B-α
        end while
        P*_i ← paretoSort(P_i)
        P_a ← P_a∪ P*_i
    end for
    P*_a ←paretoSort(P_a)
    //return Pareto Solution
    return P*_a
end procedure
```

Table 3: The characteristics of problem instances from Carter, ITC 2007 and Yeditepe benchmark datasets

	Instance	No. of Exams	No. of Students	No. of Enrolments	Conflict Density	Days	Tot. Room Capacity	No. of Cohorts
Carter	CAR91	682	16925	56877	0.13	35	682	3
	CAR92	543	18419	55522	0.14	32	543	3
	EAR83	190	1125	8109	0.27	24	190	3
	HEC92	81	2823	10632	0.42	18	81	3
	KFU93	461	5349	25113	0.06	20	461	3
	LSE91	381	2726	10918	0.06	18	381	3
	PUR93	2419	30029	120681	0.03	42	2419	3
	RYE92	486	11483	45051	0.07	23	486	3
	STA83	139	611	5751	0.14	13	139	3
	TRE92	261	4360	14901	0.18	23	261	3
	UTA92	622	21266	58979	0.13	35	622	3
	UTE92	184	2749	11793	0.08	10	184	3
	YOR83	181	941	6034	0.29	21	181	3
ITC 2007	EXAM1	607	7891	32380	0.05	54	802	3
	EXAM2	870	12743	37379	0.01	40	4076	3
	EXAM3	934	16439	61150	0.03	36	5212	3
	EXAM4	273	5045	21740	0.15	21	1200	3
	EXAM5	1018	9253	34196	0.01	42	2395	3
	EXAM6	242	7909	18466	0.06	16	2050	3
	EXAM7	1096	14676	45493	0.02	80	2530	3
	EXAM8	598	7718	31374	0.05	80	922	3
	EXAM9	169	655	2532	0.08	25	170	3
	EXAM10	214	1577	7853	0.05	32	1914	3
	EXAM11	934	16439	61150	0.03	26	4924	3
	EXAM12	78	1653	3685	0.18	12	1525	3
Yeditepe	yue20011	126	559	3486	0.18	6	450	4
	yue20012	141	591	3708	0.18	6	450	4
	yue20013	26	234	447	0.25	2	150	4
	yue20021	162	826	5755	0.18	7	550	5
	yue20022	182	869	5687	0.17	7	550	6
	yue20023	38	420	790	0.2	2	150	6
	yue20031	174	1125	6714	0.15	6	550	6
	yue20032	210	1185	6833	0.14	6	550	6

course and year. All problem instances used in this study can be downloaded from [68].

Regarding the algorithm parameter setting, only the decay rate α is required to be set up. The decay rate is set to `0.9999995` as suggested in [69]. The proposed approach was implemented in Java operating under Windows 7. All experiments were run on an Intel(R) Core(TM)i7-3820 computer with a 3.60 GHz CPU and 16.0 GB of RAM.

6.2. Experimental Results - single standard objective

Overall, the aims of the experiments in this study are two-fold; to examine the proposed approach (see Algorithm 1) over the standard benchmark single objective examination timetabling problem and evaluate the proposed approaches in enforcing fairness to determine whether fairer viable solutions exist.

From the experiments over the standard benchmark examination timetabling problems, very competitive results were obtained. The comparison between our results and recently reported results from the scientific literature is given in Table 4. As shown by Table 4 our proposed hyper-heuristic outperforms the other approaches for 8 out of 13 problem instances of the Carter dataset, 3 out of 12 problem instances of the ITC 2007 dataset, and 7 out of 8 problem instances of the Yeditepe dataset. The results also indicate that our proposed hyper-heuristic is generic, since it performs generally well over three different problem instances. In comparison, though Muller's approach[70] performs well in ITC 2007 problem instances, it underperforms when applied to the Yeditepe dataset.

Table 4: The experimental result of Algorithm 1 over three different standard benchmark examination timetabling problem datasets with their original single objective function (21 timed runs of 360 seconds each) compared with the best result reported in prior recent studies (NA: Not Applicable)

Instance	Our Result		Prior Reported Results (Best)			
	MEDIAN	BEST	(Burke,2012) [29]	(Sabar,2012) [71]	(Rahman,2014) [32]	(Burke,2014) [72]
CAR91	5.41	5.30	**5.03**	5.14	5.12	5.19
CAR92	4.62	4.51	**4.22**	4.7	4.41	4.31
EAR83	38.23	36.73	**36.06**	37.86	36.91	35.79
HEC92	11.35	**10.91**	11.71	11.9	11.31	11.19
KFU93	15.12	**14.36**	16.02	15.3	14.75	14.51
LSE91	12.09	**11.02**	11.15	12.33	11.41	10.92
PUR93	5.22	**5.03**	NA	5.37	5.87	NA
RYE92	9.58	**9.01**	9.42	10.71	9.61	NA
STA83	157.32	**157.12**	158.86	160.12	157.52	157.18
TRE92	9.13	8.75	8.37	**8.32**	8.76	8.49
UTA92	3.72	3.60	3.37	3.88	3.54	**3.44**
UTE92	26.4	**25.20**	27.99	32.67	26.25	26.7
YOR83	39.56	**38.03**	39.53	40.53	39.67	39.47
			(Muller,2007) [70]	(Sabar,2012) [71]	(Rahman,2014) [32]	(Burke,2014) [72]
EXAM1	7176	6856	4370	6234	5231	6235
EXAM2	724	632	400	**395**	433	2974
EXAM3	12429	11659	10049	13002	**9265**	15832
EXAM4	18991	**16325**	18141	17940	17,787	35106
EXAM5	4050	3837	**2988**	3900	3083	4873
EXAM6	28250	27370	26585	27000	**26,060**	31756
EXAM7	5848	5528	**4213**	6214	10,712	11562
EXAM8	10178	9798	**7742**	8552	12,713	20994
EXAM9	1320	1246	**1030**	NA	1111	NA
EXAM10	15239	**14556**	16682	NA	14,825	NA
EXAM11	40109	36810	**34129**	NA	28,891	NA
EXAM12	5581	**5300**	5535	NA	6181	NA
			(Muller,2007) [70]	(Sabar,2012) [71]	(Rahman,2014) [32]	(Burke,2014) [72]
yue20011	68	**56**	62	NA	NA	NA
yue20012	161	**122**	125	NA	NA	NA
yue20013	29	**29**	29	NA	NA	NA
yue20021	111	76	**70**	NA	NA	NA
yue20022	212	**162**	170	NA	NA	NA
yue20023	61	**56**	70	NA	NA	NA
yue20031	206	**143**	223	NA	NA	NA
yue20032	479	**434**	440	NA	NA	NA

6.3. Experimental Results - standard objective and fairness

With the aim of improving fairness, in addition to standard single objective function, we tested two different methods. First, the single objective approach (see Algorithm 1) as discussed in Section 5, in which we simply change the objective function to maximise fairness, i.e. maximise $JFI(A)$ (see Equation 5), and so minimise unfairness $AJFI(A)$, replacing the standard objective function. We add 'not worsening the standard objective function' as a hard constraint. The second method involves the scalarisation approach as shown in Algorithm 2. In this experiment, instead of generating an initial solution from scratch, the best solutions from Table 4) were used as initial solutions.

Our experimental results of the first approach showed that the fairness of solutions had minimal improvement. Only 7 out of 33 instances became very slightly (less than 0.5%)fairer without making the standard objective function worse. This indicates that in the majority of instances, there is trade-off between the standard penalty and the fairness objective function, in that improving one objective can degrade the other.

For the second approach, we ran Algorithm 2 21 times, taking less than 1 minute per run (set `N=21 and T2=60000`). We tested over both the bi-objective and the three-objective problems. The experimental results are presented in Table 5.

The value range (represented as min-max values) of the two objectives of the final Pareto set of solutions obtained during the experiments is provided in Table 5. We observe that the solutions for all instances achieved increased fairness while only slightly compromising the standard penalty.

To illustrate the trade-off between the two objectives, i.e. standard penalty and unfairness as defined in Equation 14, one instance was chosen from each of the benchmark datasets. These were HEC-92, STA83, EXAM4, and yue20011, a sample of those for which our proposed algorithm achieved better results than reported in the literature (see Table 4). As with previous experimentation, the solver was run 21 times for each dataset, but allowed 360 seconds instead of the previously allotted 60 seconds running-time. Figure 2 illustrates the solutions in the Pareto set achieved by the proposed approach for the instances HEC92, STA83, EXAM4, and yue20011.

As shown in Figure 2, in terms of the first objective function, i.e. the standard penalty, the values for the problem instances HEC92, STA83, EXAM4, YUE20011 range between 10.91-16.84, 157.12-172.83, 16324-37761, and 54-1055, respectively. Similarly, in terms of the second objective function, i.e. unfairness that is measured by AJFI(A), the values range between 0.37-0.51, 0.05-0.10, 0.37-0.71, and 0.03-0.92.

At the extreme point of the Pareto set of solutions, for the `STA83` problem instance, we can improve the cohort fairness by about `5%` (from `0.10` to `0.05`) with the effect of worsening the standard penalty by about `10%` (from `157.12` to `172.83`). On average, improving fairness by `1.35%` resulted with a worsening of `1.93%` of the standard penalty. The final policy decision on the trade-off between the two objectives is up to the decision maker. For instance, the decision maker may cap any degradation of the standard penalty to a maximum of `3%`.

Table 5: The objective function values range (i.e. the values range between min and max) of the final Pareto set of solutions with two- and three-objective functions: standard objective function, i.e. std.penalty, overall unfairness, i.e. AJFI(A), and unfairness within a cohort, i.e, AJFI(C) using the proposed hyper-heuristic approach

	Two Objectives				Three Objectives					
	Std.P		AJFI(A)		Std.P		AJFI(A)		AJFI(C)	
Instance	min	max	min	max	min	max	min	max	min	max
CAR91	5.30	6.56	0.63	0.68	5.30	8.45	0.61	0.68	0.61	0.68
CAR92	4.51	5.70	0.68	0.71	4.51	7.84	0.64	0.71	0.64	0.71
EAR83	36.73	50.33	0.14	0.17	36.71	50.29	0.14	0.17	0.13	0.17
HEC92	10.91	16.84	0.37	0.51	10.91	22.17	0.35	0.51	0.35	0.51
KFU93	14.36	22.31	0.28	0.45	14.36	22.33	0.28	0.45	0.27	0.45
LSE91	11.01	16.28	0.36	0.49	11.02	16.28	0.36	0.49	0.35	0.49
PUR93	5.03	6.00	0.64	0.67	5.03	7.44	0.62	0.67	0.62	0.67
RYE92	9.01	15.35	0.53	0.63	9.01	9.01	0.53	0.53	0.53	0.53
STA83	157.12	172.81	0.05	0.10	157.12	196.61	0.03	0.10	0.01	0.10
TRE92	8.74	10.27	0.53	0.56	8.75	14.38	0.49	0.56	0.49	0.56
UTA92	3.60	5.53	0.72	0.77	3.60	6.38	0.70	0.77	0.70	0.77
UTE92	25.20	43.27	0.17	0.21	25.20	43.26	0.17	0.21	0.16	0.21
YOR83	38.03	44.37	0.22	0.25	38.03	44.38	0.22	0.25	0.22	0.25
EXAM1	6855	8312	0.54	0.60	6855	20662	0.52	0.60	0.51	0.60
EXAM2	632	932	0.85	0.92	632	932	0.85	0.98	0.36	0.98
EXAM3	11653	34021	0.84	0.91	11653	59930	0.78	0.91	0.78	0.91
EXAM4	16325	37264	0.37	0.71	16325	59406	0.31	0.71	0.29	0.71
EXAM5	3837	5434	0.50	0.64	3837	37430	0.03	0.64	0.03	0.64
EXAM6	27370	38550	0.74	0.76	27370	63055	0.69	0.76	0.68	0.76
EXAM7	5528	9828	0.55	0.79	5528	12122	0.44	0.79	0.43	0.79
EXAM8	9787	10216	0.56	0.58	9794	13005	0.48	0.58	0.46	0.58
EXAM9	1225	1723	0.48	0.64	1245	1763	0.47	0.63	0.46	0.63
EXAM10	14556	15941	0.59	0.63	14556	72129	0.50	0.63	0.49	0.63
EXAM11	36809	55821	0.81	0.89	36810	167640	0.77	0.89	0.77	0.89
EXAM12	5286	12076	0.80	0.88	5288	12038	0.80	0.88	0.80	0.88
yue20011	54	1054	0.03	0.92	54	1054	0.03	0.92	0.03	0.92
yue20012	118	1118	0.07	0.86	119	85824	0.00	0.86	0.00	0.86
yue20013	29	95	0.65	0.88	29	104	0.56	0.88	0.56	0.88
yue20021	76	1076	0.05	0.92	76	87934	0.00	0.92	0.00	0.92
yue20022	160	1181	0.11	0.87	157	96071	0.00	0.87	0.00	0.87
yue20023	56	143	0.69	0.89	56	161	0.63	0.89	0.63	0.89
yue20031	142	1142	0.13	0.90	142	94160	0.00	0.90	0.00	0.90
yue20032	434	1517	0.22	0.77	434	1517	0.22	0.77	0.22	0.77

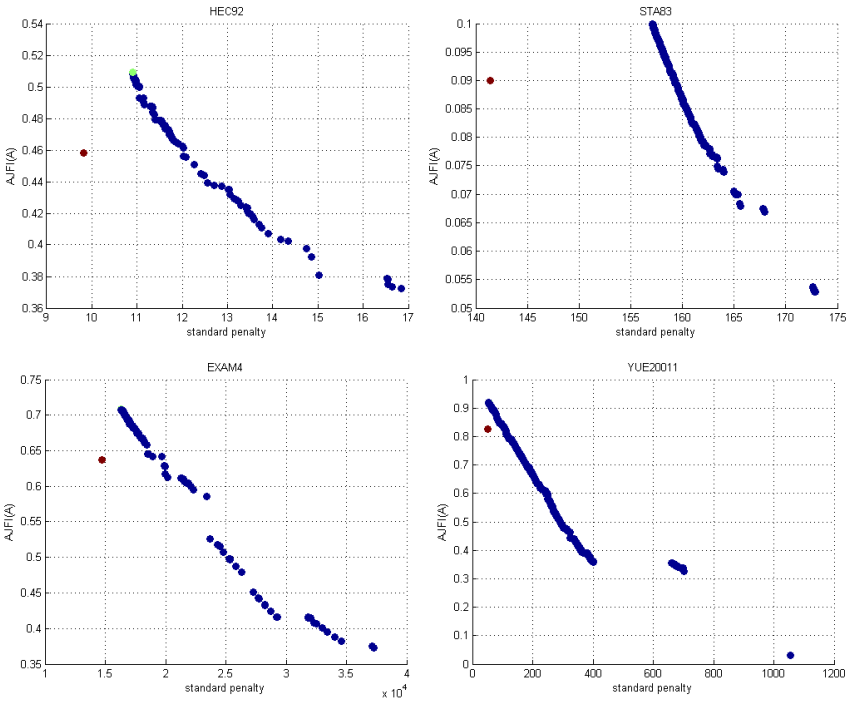

Figure 2: The final Pareto set of solutions with two objectives: standard penalty and overall unfairness, i.e. AJFI(A), for instances HEC92, STA83, EXAM4, and YUE20011. The red point is the reference point and the green point is the initial solution.

6.3.1. Experimental Result on Enforcing Fairness "Within a cohort"

In order to enforce fairness "within a cohort" (see Section 4), a new objective function is introduced. Thus, there are three objective functions that have to be optimised, i.e. minimising standard quality of the solution (see Equation 1 and 4), maximising overall fairness (i.e. Equation 5), and maximising fairness within a cohort (i.e. Equation 6). As discussed in the previous section, the maximisation problems are then changed to minimisation problems. Thus the objective functions would be standard penalty, AJFI(A), and AJFI(C).

In addition, a new problem data set was created in order to allow preliminary experimentation with this alternative definition of fairness, as in "Fairness within a cohort". The existing problem instances of either Carter or ITC 2007 datasets do not specify the course for each student, while the Yeditepe dataset instances contain information about each individual student's course and year of admission. Therefore, in our experiment the cohort for the Yeditepe dataset was based on the course for each student, i.e students within the same course are considered to be within the same cohort. For the other problem instances from Carter and ITC Dataset we clustered students based on the exams enrolled by the students using machine learning technique.

Given three objective functions, the experimentation was conducted in exactly the same manner as when generating a Pareto set of solutions with two objectives.

To illustrate the trade-off between standard penalty, overall fairness, and average fairness within a cohort, Figure 3 visualises the final Pareto set of solutions in "parallel coordinates"[73] generated by using the proposed approach. To make the visualisation more readable, we filtered the Pareto set of solutions with standard objective function values less than `158`.

From the visualisation, we can observe that there is obvious inverse-correlation between the standard penalty and overall unfairness, AJFI(A). In this sense, decreasing the standard penalty will increase unfairness. However, the correlation between overall unfairness and unfairness within a cohort is not quite as obvious. The user or decision maker will most probably prefer a solution with a standard penalty slightly worse than the best, has reasonable overall fairness, but still has very good fairness within a cohort. An example of such a solution is indicated with solution `74` in Figure 3 and Figure 4. The value of each objective function is given in Table 7 while the changes of its objective function is given in Table 8. Finally, how the solutions affect students is visualised by Figure 6. The very existence of such solutions (fairer timetables) is an important contribution of this work. We expect that improved future algorithms, better tailored to fairness measures, should make it easier to find them.

Of course, if the decision maker is much more concerned about fairness within a cohort as opposed to overall fairness, they can just focus on making a trade-off between the standard penalty and fairness as shown by Figure 5. The figure visualises the final Pareto set of solutions with two objectives, i.e. standard penalty and unfairness within a cohort. Table 6 presents the objective function values of the numbered solutions in Figure 5. The table also presents

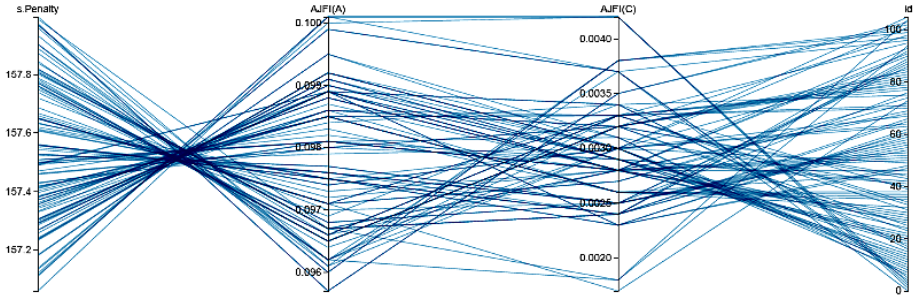

Figure 3: Final Pareto set of solutions for instance STA-83 generated with the proposed hyper-heuristics approach represented in parallel coordinate: Trade-off between standard penalty, overall unfairness i.e. AJFI(A), and average unfairness within a cohort i.e. AJFI(C)

the percentage of change in objective function values (i.e. delta) from solution `1` to the other selected solutions.

Table 6: Objective function values of the solutions visualised in Figure 5

	O. Funct. Values		Delta (%)	
sol#	s.Pen(A)	AJFI(C)	s.Pen(A)	AJFI(C)
1	157.12	0.00635	-	-
2	158.80	0.00560	1.07	-11.75
3	162.14	0.00521	3.19	-17.93
4	167.09	0.00467	6.34	-26.45
5	172.01	0.00409	9.47	-35.64

In Figure 5, the left two points (solution `1` and `2`) show that the cohort unfairness can be decreased significantly from `0.00635` to `0.00560`, or about `11.75%`, by just increasing the standard penalty from `157.12` to `158.80`, about `1.07%`.

The gain is much larger that can be obtained in the overall fairness, in Figure 2. This makes sense as seen in Figure 6(a), for a best-standard solution, the 3 cohorts has very different average penalties and so there is not much that can be done to improve fairness. However, in cohort 3, there are two distinct groups of students, with different penalties. Figure 6(b) shows a solution with a weight applied in order to reduce the cohort unfairness and where the two groups in that cohort end up with closer penalty values.

Table 7: The objective function values of the selected non-dominated solutions: sol 1, 74 and 103

Sol ID	s.Pen (A)	AJFI(A)	s.Pen(C1)	AJFI(C1)	s.Pen(C2)	AJFI(C2)	s.Pen(C3)	AJFI(C3)	AJFI(C)
1	157.06	0.1001	226.00	0	126.38	0.0097	136.87	0.0014	0.0037
74	157.66	0.0971	226.04	0	127.62	0.0038	137.28	0.0014	0.0017
103	157.97	0.0957	226.00	0	128.96	0.0086	136.95	0.0019	0.0035

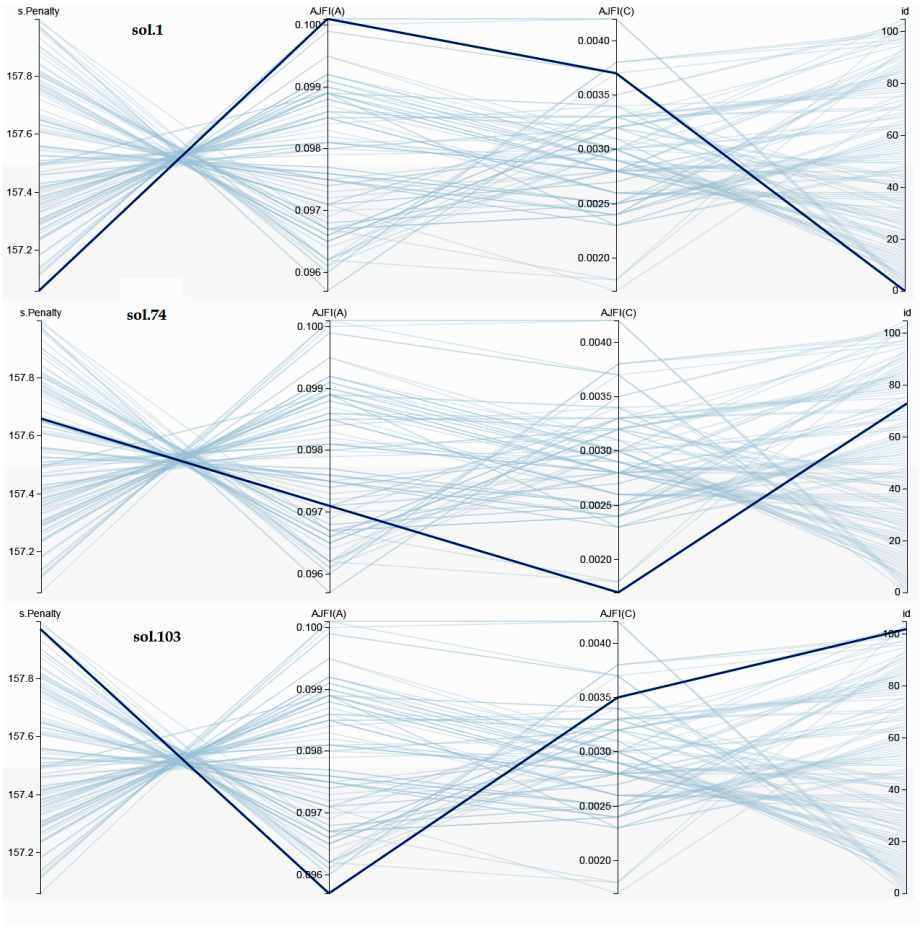

Figure 4: Three selected solutions (id:1,74,103) from Final Pareto set of solutions for instance STA-83 generated with the proposed hyper-heuristics approach represented in parallel coordinate: Trade-off between standard penalty, overall unfairness i.e. AJFI(A), and average unfairness within a cohort i.e. AJFI(C)

Table 8: The changes (in percentage) of objective function values if the selected non-dominated solutions: sol 1, 74 and 103 are compared to each other

Sol #	1			74			103		
	s.Pen (A)	AJFI(A)	AJFI(C)	s.Pen (A)	AJFI(A)	AJFI(C)	s.Pen (A)	AJFI(A)	AJFI(C)
1	**X**	**X**	**X**	0.38	-3.00	-54.05	0.58	-4.40	-5.41
74	-0.38	3.09	117.65	**X**	**X**	**X**	0.20	-1.44	105.88
103	-0.58	4.60	5.71	-0.20	1.46	-51.43	**X**	**X**	**X**

The objective function values of the solutions visualised in Figure 4 and Figure 6 are given in Table 7, with the differences in these objective function values over each solution presented in Table 8. For example, from solution `1` to solution `74`, we can decrease 'unfairness within cohort' by `54.05%` as a

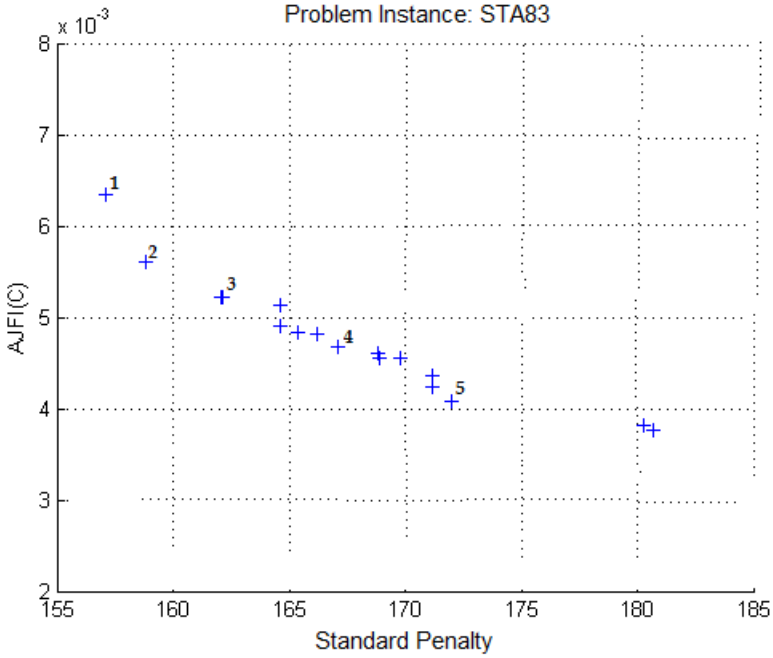

Figure 5: The final Pareto set of solutions with two objectives: standard penalty and unfairness within cohort, i.e. AJFI(C), for instance STA83.

consequence of increasing the standard penalty by 0.38%.

7. Conclusion

Our survey of student views found that over half of them were unhappy with their examination timetables. Furthermore, about 30% of respondents even believed that their examination timetable negatively affected their academic achievement. We have no evidence that the timetables actually did affect student performance, but the perception is important; especially with Universities competing for students. Therefore, this work intends to contribute to generating examination timetables that match student preferences and enhance their satisfaction. In particular, we have proposed and studied methods to improve fairness amongst students in the timetables they receive. A crucial contribution of this paper is to introduce the novel concept of 'fairness within a cohort of students'; this complements and widens the concept of fairness within the entire student body. To support this, we proposed a specific formulation of these concepts with an associated algorithm, based on hyper-heuristics, together with a multi-objective optimisation approach to improve fairness. We have presented experimental results showing that, unsurprisingly, there is a non-trivial Pareto Front; in other words, there exists a trade-off between enhancing fairness and satisfying the standard objective function. It is possible to improve

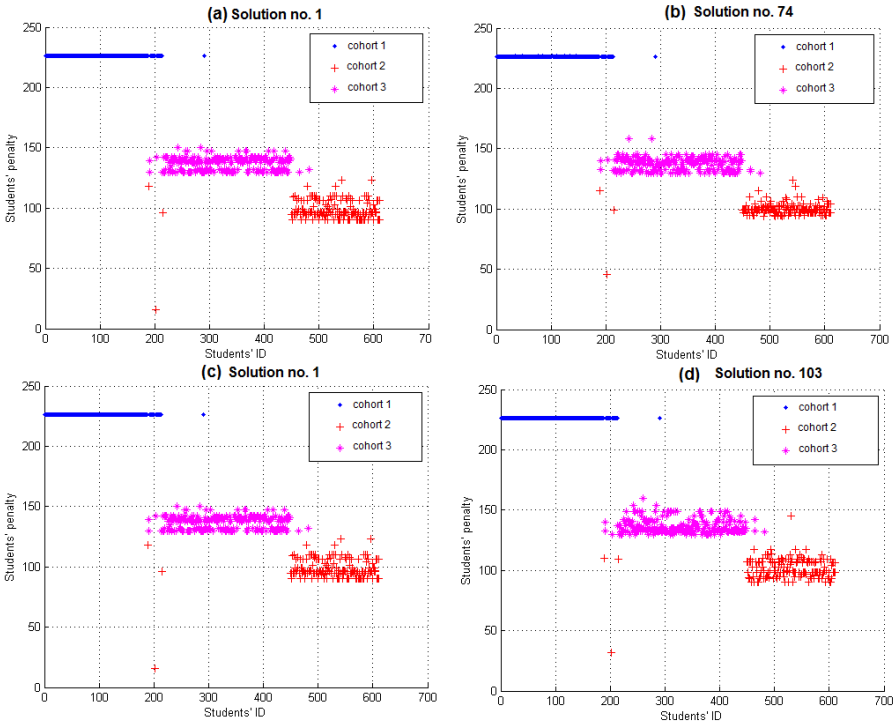

Figure 6: Penalty associated with each student within three non-dominated solutions: Sol.1, sol.74, and sol.103 for problem instance STA83. A different colour represents a different cohort.

fairness overall and within cohorts, though, of course, this results in slightly increasing the standard soft constraints violation penalty. Future work should investigate whether the fairer timetables are in practice actually preferred by students; such studies may well further refine the notions of fairness. Also, although we use fairness measures based on the Jain fairness index (JFI), we are not claiming that such JFI-based measures are the only reasonable ones. Other formulations could be studied for fairness, such as GINI index, or the simple application of higher powers than the quadratic implicit in the JFI measure (e.g. see our preliminary work in [5]). Also, although we have used stochastic local search methods, for small problems it may be feasible to use exact integer programming methods, possibly in the form of non-linear extensions along the lines of branch-and-cut in [74]. Of course, many other meta-heuristics may by applicable.

As a final but important note regarding fairness within 'cohort' information, we observe that current studies are somewhat hampered because the existing benchmarks do not include the 'meta-data' (e.g. information about student's course and year, exam's school and faculty) that can be used to define 'cohorts'. Hence, we strongly encourage researchers and practitioners in the area, and all

those who create and share public datasets, to also preserve and share suitable meta-data. Such meta-data can then be used to aid the development of formulations and algorithms that better meet student preferences.

Acknowledgement

This study was supported by Directorate General of Higher Education (DGHE), Ministry of Research, Technology, and Higher Education of Indonesia.

References

[1] M. W. Carter, G. Laporte, S. Y. Lee, Examination timetabling: Algorithmic strategies and applications, Journal of the Operational Research Society 47 (3) (1996) 373–383.

[2] E. Özcan, E. Ersoy, Final exam scheduler - FES, in: Proceedings of the 2005 IEEE Congress on Evolutionary Computation, Vol. 2, 2005, pp. 1356–1363.

[3] B. McCollum, P. McMullan, A. J. Parkes, E. K. Burke, R. Qu, A new model for automated examination timetabling, Annals of Operations Research 194 (1) (2012) 291–315.

[4] B. McCollum, A. Schaerf, B. Paechter, P. McMullan, R. Lewis, A. J. Parkes, L. D. Gaspero, R. Qu, E. K. Burke, Setting the research agenda in automated timetabling: The second international timetabling competition, INFORMS Journal on Computing 22 (1) (2010) 120–130.

[5] A. Muklason, A. J. Parkes, B. McCollum, E. Özcan, Initial results on fairness in examination timetabling, in: Proceedings of the 6th Multidisciplinary International Conference on Scheduling : Theory and Applications (MISTA), 2013, pp. 777–780.

[6] A. Muklason, A. J. Parkes, B. McCollum, E. Özcan, Fairness in examination timetabling problems: A survey and the new problem formulation, in: Proceedings of the 10th International Conference on The Practice and Theory of Automated Timetabling (PATAT 2014), 2014, pp. 512–515.

[7] M. Mühlenthaler, R. Wanka, Fairness in academic course timetabling, Annals of Operations Research.

[8] A. Schaerf, A survey of automated timetabling, Artificial Intelligence Review 13 (2) (1999) 87–127.

[9] E. Burke, D. Elliman, P. Ford, R. Weare, Examination timetabling in British universities: A survey, in: Practice and Theory of Automated Timetabling, Vol. 1153 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 1996, pp. 76–90.

[10] B. Mccollum, P. Mcmullan, E. K. Burke, A. J. Parkes, R. Qu, The second international timetabling competition: Examination timetabling track (2007).

[11] E. K. Burke, J. P. Newall, R. F. Weare, A memetic algorithm for university exam timetabling, in: E. Burke, P. Ross (Eds.), Practice and Theory of Automated Timetabling, Vol. 1153 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 1996, pp. 241–250.

[12] L. T. Merlot, N. Boland, B. D. Hughes, P. J. Stuckey, A hybrid algorithm for the examination timetabling problem, in: E. Burke, P. De Causmaecker (Eds.), Practice and Theory of Automated Timetabling IV, Vol. 2740 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2003, pp. 207–231.

[13] G. Kendall, N. Hussin, A tabu search hyper-heuristic approach to the examination timetabling problem at the MARA University of Technology, in: E. Burke, M. Trick (Eds.), Practice and Theory of Automated Timetabling V, Vol. 3616 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2005, pp. 270–293.

[14] M. Ayob, S. Abdullah, A. M. A. Malik, A practical examination timetabling problem at the Universiti Kebangsaan Malaysia, International Journal of Computer Science and Network Security 7 (9) (2007) 198–204.

[15] A. J. Parkes, E. Özcan, Properties of Yeditepe examination timetabling benchmark instances, in: Proceedings of the 8th International Conference on Practice and Theory of Automated Timetabling (PATAT), 2010, pp. 531–534.

[16] M. Kahar, G. Kendall, The examination timetabling problem at Universiti Malaysia Pahang: comparison of a constructive heuristic with an existing software solution, European Journal of Operational Research 207 (2) (2010) 557 – 565.

[17] P. Demeester, B. Bilgin, P. De Causmaecker, G. Vanden Berghe, A hyper-heuristic approach to examination timetabling problems: benchmarks and a new problem from practice, Journal of Scheduling 15 (1) (2012) 83–103.

[18] M. W. Carter, A survey of practical applications of examination timetabling algorithms, Operations Research 34 (2) (1986) 193–202.

[19] E. K. Burke, S. Petrovic, Recent research directions in automated timetabling, European Journal of Operational Research 140 (2) (2002) 266 – 280.

[20] R. Qu, E. K. Burke, B. McCollum, L. T. Merlot, S. Y. Lee, A survey of search methodologies and automated system development for examination timetabling, Journal of Scheduling 12 (1) (2009) 55–89.

[21] E. K. Burke, M. Gendreau, M. Hyde, G. Kendall, G. Ochoa, E. Özcan, R. Qu, Hyper-heuristics: a survey of the state of the art, J Oper Res Soc 64 (12) (2013) 1695–1724.

[22] S. Asta, E. Özcan, A. J. Parkes, Batched mode hyper-heuristics, Lecture Notes in Computer Science 7997 (2013) 404–409, proceedings of LION 2013.

[23] A. J. Parkes, E. Özcan, D. Karapetyan., A software interface for supporting the application of data science to optimisation, in: Proceedings of Learning and Intelligent Optimization Conference 2015 LION9, 2015, pp. 306–311.

[24] B. Bilgin, E. Özcan, E. E. Korkmaz, An experimental study on hyper-heuristics and exam timetabling, in: E. K. Burke, H. Rudova (Eds.), Practice and Theory of Automated Timetabling VI, Vol. 3867 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2007, Ch. 25, pp. 394–412.

[25] E. K. Burke, B. McCollum, A. Meisels, S. Petrovic, R. Qu, A graph-based hyper-heuristic for educational timetabling problems, European Journal of Operational Research 176 (1) (2007) 177 – 192.

[26] E. Özcan, Y. Bykov, M. Birben, E. K. Burke, Examination timetabling using late acceptance hyper-heuristics, in: Proceedings of the IEEE Congress on Evolutionary Computation CEC '09, 2009, pp. 997–1004.

[27] N. Pillay, Evolving hyper-heuristics for the uncapacitated examination timetabling problem, Journal of the Operational Research Society 63 (1) (2011) 47–58.

[28] E. Burke, G. Kendall, M. Misir, E. Özcan, Monte carlo hyper-heuristics for examination timetabling, Annals of Operations Research 196 (1) (2012) 73–90.

[29] E. K. Burke, N. Pham, R. Qu, J. Yellen, Linear combinations of heuristics for examination timetabling, Annals of Operations Research 194 (1) (2012) 89–109.

[30] A. Soghier, R. Qu, Adaptive selection of heuristics for assigning time slots and rooms in exam timetables, Applied Intelligence 39 (2) (2013) 438–450.

[31] S. Abdul-Rahman, E. Burke, A. Bargiela, B. McCollum, E. Özcan, A constructive approach to examination timetabling based on adaptive decomposition and ordering, Annals of Operations Research 218 (1) (2014) 3–21.

[32] S. Abdul Rahman, A. Bargiela, E. K. Burke, E. Özcan, B. McCollum, P. McMullan, Adaptive linear combination of heuristic orderings in constructing examination timetables, European Journal of Operational Research 232 (2) (2014) 287–297.

[33] N. R. Sabar, M. Ayob, G. Kendall, R. Qu, A dynamic multiarmed bandit-gene expression programming hyper-heuristic for combinatorial optimization problems, IEEE Transactions on Cybernetics 45 (2) (2015) 217–228.

[34] M. Ayob, G. Kendall, A Monte Carlo hyper-heuristic to optimise component placement sequencing for multi head placement machine, in: Proceedings of the Placement Machine, INTECH03, Thailand, 2003, pp. 132–141.

[35] R. Qu, E. K. Burke, Hybridizations within a graph-based hyper-heuristic framework for university timetabling problems, J Oper Res Soc 60 (9) (2008) 1273–1285.

[36] D. E. Joslin, D. P. Clements, "Squeaky Wheel" optimization, Journal of Articial Intelligence Research 10 (1999) 353–373.

[37] D. K. Foley, Resource allocation in the public sector, Yale Economic Essays 7 (1) (1967) 45–98.

[38] E. A. Pazner, Pitfalls in the theory of fairness, Journal of Economic Theory 14 (1977) 458–466.

[39] R. Denda, A. Banchs, W. Effelsberg, The fairness challenge in computer networks, in: Proceedings of the Quality of Future Internet Services, Springer, 2000, pp. 208–220.

[40] E. Altman, K. Avrachenkov, S. Ramanath, Multi-scale fairness and its application to resource allocation in wireless networks, Computer Communications 35 (7) (2012) 820 – 828.

[41] D. J. Leith, Q. Cao, V. G. Subramanian, Max-min fairness in 802.11 mesh networks, IEEE/ACM Transactions on Networking, 20 (3) (2012) 756–769.

[42] K. Ronasi, V. W. S. Wong, S. Gopalakrishnan, Distributed scheduling in multihop wireless networks with maxmin fairness provisioning, IEEE Transactions on Wireless Communications 11 (5) (2012) 1753–1763.

[43] L. Zhang, W. Luo, S. Chen, Y. Jian, End-to-end maxmin fairness in multihop wireless networks: Theory and protocol, Journal of Parallel and Distributed Computing 72 (3) (2012) 462–474.

[44] J. M. Jaffe, Bottleneck Flow Control, IEEE Transactions on Communications 29 (7).

[45] J. Mo, J. Walrand, Fair end-to-end window-based congestion control, IEEE/ACM Transactions on Networking 8 (5) (2000) 556–567.

[46] M. J. Soomer, G. M. Koole, Fairness in the aircraft landing problem, in: Anna Valicek Competition 2008, 2008.

[47] Y. Wang, F. Wang, D. Wang, Z. Gong, J. Liu, Revisit the fairness issues in flight landing scheduling, in: Proceedings of the 15th International IEEE Conference on Intelligent Transportation Systems (ITSC), 2012, pp. 1435–1440.

[48] M. H. A. Ibrahim, R. Ahmad, N. K. Ibrahim, S. Chuprat, H. Haron, Nurse scheduling with fairness criteria for public hospital, in: Proceedings of the IEEE International Conference on Computer Applications and Industrial Electronics (ICCAIE), 2011, pp. 91–95.

[49] P. Smet, S. Martin, O. Djamila, E. Özcan, G. Vanden Berghe, Investigation of fairness measures for nurse rostering, in: Proceedings of the 9th International Conference on the Practice and Theory of Timetabling (PATAT), 2012, pp. 369–372.

[50] P. Smet, B. Bilgin, P. De Causmaecker, G. Vanden Berghe, Modelling and evaluation issues in nurse rostering, Annals of Operations Research 218 (1) (2014) 303–326.

[51] S. Martin, D. Ouelhadj, P. Smet, G. V. Berghe, E. Özcan, Cooperative search for fair nurse rosters, Expert Systems with Applications 40 (16) (2013) 6674 – 6683.

[52] C. Castro, S. Manzano, Variable and value ordering when solving balanced academic curriculum problems, in: Proceedings of the 6th Workshop of the ERCIM WG on Constraints, 2001.

[53] L. D. Gaspero, A. Schaerf, Hybrid local search techniques for the generalized balanced academic curriculum problem, in: Proceedings of the 5th International Workshop on Hybrid Metaheuristics, HM '08, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 146–157.

[54] M. Chiarandini, L. Di Gaspero, S. Gualandi, A. Schaerf, The balanced academic curriculum problem revisited, Journal of Heuristics 18 (1) (2012) 119–148.

[55] M. Mühlenthaler, R. Wanka, Fairness in academic timetabling, in: Proceedings of the 9th Int. Conf. on the Practice and Theory of Automated Timetabling(PATAT), 2012, pp. 114–130.

[56] D. Bertsekas, R. Gallager, Data Networks, Prentice-Hall, Inc., New Jersey, USA, 1992.

[57] R. K. Jain, D.-M. W. Chiu, W. R. Hawe, A quantitative measure of fairness and discrimination for resource allocation in shared computer system (1984).

[58] A. Bonutti, F. De Cesco, L. Di Gaspero, A. Schaerf, Benchmarking curriculum-based course timetabling: formulations, data formats, instances, validation, visualization, and results, Annals of Operations Research 194 (1) (2012) 59–70.

[59] P. Cowling, G. Kendall, N. M. Hussin, A survey and case study of practical examination timetabling problems, in: Proceedings of the 4th International Conference on the Practice and Theory of Automated Timetabling PATAT02, 2002, pp. 258–261.

[60] S. Rahman, A. Bargiela, E. Burke, E. Ozcan, B. McCollum, Construction of examination timetables based on ordering heuristics, in: Proceedings of the 24th International Symposium on Computer and Information Sciences, 2009. ISCIS 2009., 2009, pp. 680–685.

[61] E. Özcan, M. Misir, G. Ochoa, E. K. Burke, A reinforcement learning - great-deluge hyper-heuristic for examination timetabling, International Journal of Applied Metaheuristic Computing (IJAMC) 1 (1) (2010) 39–59.

[62] J. M. Thompson, K. A. Dowsland, A robust simulated annealing based examination timetabling system, Computers & Operations Research 25 (7-8) (1998) 637–648.

[63] G. Ochoa, M. Hyde, T. Curtois, J. A. Vazquez-Rodriguez, J. Walker, M. Gendreau, G. Kendall, B. McCollum, A. J. Parkes, S. Petrovic, E. K. Burke, Hyflex: A benchmark framework for cross-domain heuristic search, in: J.-K. Hao, M. Middendorf (Eds.), Evolutionary Computation in Combinatorial Optimization, Vol. 7245 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2012, pp. 136–147.

[64] S. Asta, E. Özcan, A. J. Parkes, Batched mode hyper-heuristics, in: G. Nicosia, P. Pardalos (Eds.), Learning and Intelligent Optimization, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2013, pp. 404–409.

[65] M. Kahar, G. Kendall, A great deluge algorithm for a real-world examination timetabling problem, Journal Of The Operational Research Society.

[66] K. Miettinen, Nonlinear Multi-objective Optimization, Kluwer, Boston, MA, 1999, 1999.

[67] H. T. Kung, F. Luccio, F. P. Preparata, On finding the maxima of a set of vectors, J. ACM 22 (4) (1975) 469–476.

[68] A. Muklason, Exam Timetabling Problem With Fairness Datasets (2015 (accessed February 1, 2015)).
URL `http://www.cs.nott.ac.uk/~abm/research/exam-fairness-data/examproblemwithfairness/`

[69] A. Hmer, M. Mouhoub, A multi-phase hybrid metaheuristics approach for the exam timetabling, in: Proceeding of 10th International Conference of the Practice and Theory of Automated Timetabling PATAT 2014, 26-29 August 2014, York, United Kingdom, 2014, pp. 233–251.

[70] T. Müller, ITC2007 solver description: a hybrid approach, Annals of Operations Research 172 (1) (2009) 429–446.

[71] N. R. Sabar, M. Ayob, R. Qu, G. Kendall, A graph coloring constructive hyper-heuristic for examination timetabling problems, Applied Intelligence 37 (1) (2012) 1–11.

[72] E. K. Burke, R. Qu, A. Soghier, Adaptive selection of heuristics for improving exam timetables, Annals of Operations Research 218 (1) (2014) 129–145.

[73] A. Inselberg, Parallel Coordinates: Visual Multidimensional Geometry and Its Applications, 1st Edition, Springer, 2009.

[74] E. K. Burke, J. Mareček, A. J. Parkes, H. Rudová, A branch-and-cut procedure for the Udine course timetabling problem, Annals of Operations Research 194 (1) (2012) 71–87.