# Analysing Similarity in Examination Timetabling

E. K. Burke[1], A. J. Eckersley[1], B. McCollum[2], S. Petrovic[1], and R. Qu[1]

[1] University of Nottingham, Nottingham, NG8 1BB, UK
{ekb, aje, sxp, rxq}@cs.nott.ac.uk,
[2] Queen's University of Belfast, Belfast, BT7 1NN, UK
b.mccollum@qub.ac.uk

**Abstract.** In this paper we carry out an investigation into some of the major features of exam timetabling problems with a view to developing a similarity measure. This similarity measure will be used within a case-based reasoning (CBR) system to match a new problem with one from a case-base of previously solved problems. The case base will also store the heuristic or meta-heuristic technique(s) applied most successfully to each problem stored. The technique(s) stored with the matched case will be retrieved and applied to the new case. The motivation behind our similarity measure is that *similar* problems should be solved equally well by the same technique. The aim of this paper is to discuss which features and combinations of features of exam timetabling problems are potentially important for measuring such similarity. These will be the features whose values must be similar between two problems for the whole problems to be considered as *similar*.

## 1 Introduction

In exam timetabling, a set of exams must be scheduled to a set of timeslots subject to certain constraints. In most real life problems there will also need to be some allocation of resources (e.g. rooms, invigilators and special equipment) to each exam. The constraints on the problem can be divided into *hard constraints* and *soft constraints*. Hard constraints are those which must be satisfied in order for the resulting timetable to be considered feasible. Soft constraints are those which are desirable to be satisfied, but which in general cannot all be wholly met. The quality of a resulting timetable can be measured according to an objective function which weights the violation of the different soft constraints. One aim of the timetabling problem is to create a timetable which minimises this objective function, meaning that as many soft constraints as possible are satisfied.

Both hard and soft constraints vary between institutions, as can be seen from the survey of British universities carried out by Burke et. al. [4]. The most common hard constraints can be summarised as follows:

- Every exam must be scheduled in exactly one timeslot
- Every exam must be assigned to a room(s) of sufficient size and assigned an invigilator(s)

– No student must be scheduled to be in two different exams at the same time
– There must be enough seats in each period for all exams scheduled
– Certain exams must be scheduled into specific timeslots or rooms
– Certain exams must take place simultaneously

Of course, depending on the institution in question, some of these hard constraints may not be relevant. Some universities have very tight constraints on room space whereas others have a large amount of rooms usable for exams so that the constraint on room availability is unnecessary. Soft constraints tend to be a lot more varied and much more dependent on the peculiarities of each institution since it is these which measure, amongst a given set of feasible timetables, which are the most desirable. For some exam timetabling problems it is difficult to find a feasible solution at all, whereas for other problems there are a large number of feasible solutions and the focus of the problem solving is very much directed to the minimisation of soft constraint violations. Soft constraints often encountered include the following [4]:

– Exams for each student should be spread as far apart as possible
– A student should not be required to take x exams in y periods
– Time windows for certain exams
– No more than x exams taking place simultaneously
– No more than y students scheduled to sit exams at any one time
– Exams should not be split across rooms
– No more than one exam in a room at a time
– Teacher or student preferences
– Distance between rooms holding a given exam should be minimised (when the exam is split across two or more rooms)
– The total number of periods should be minimised

Some constraints may be soft in one problem, but hard in another, depending on the requirements of the universities. In some cases, the total number of periods is fixed *a priori* and is taken as a hard constraint, whilst in others this is taken into account in the objective function with one of the aims being to minimise the length of timetable produced. It may also be the case that different institutions with the same basic constraints place differing weights on the soft constraints in the objective function to define what they consider to be a *good* timetable. It can often be the case that two or more soft constraints on a given problem are conflicting, such as the wish to spread exams as far apart as possible and the wish to minimise the total length of the timetable.

The exam timetabling problem lends itself well to being tackled by a variety of different meta-heuristic techniques. These are essentially techniques which work in the solution space of exam timetables, having been seeded with an initial solution which is often found using a fast graph colouring heuristic (e.g. [5]) which sequentially assigns each exam into an initially empty timetable until all exams are assigned to a timeslot (e.g. [8]). Such construction heuristics can themselves produce solutions of an acceptable quality and are often combined with a second, improvement phase(e.g. [21, 22, 27]).

A range of different meta-heuristic techniques have been applied to exam timetabling in recent years. Amongst those successfully applied include Simulated Annealing (e.g. [39, 40]), Tabu Search (e.g. [34, 36, 41]), Great Deluge (e.g. [12]), Fuzzy Systems (e.g. [1]) and Memetic algorithms (e.g. [7, 9]). Amongst the current methods being applied to exam timetabling are Ant Colony optimisation and Variable Neighbourhood Search. For a wider collection of examples of exam timetabling techniques, consult the Selected Papers volumes from the past PATAT conferences [30–33]. For an overview of the area, see the following survey papers [26, 19, 2, 20, 6, 35]. Burke et al. [13, 17, 38] propose a case-based reasoning (CBR) technique for selecting amongst a set of previously used heuristics and meta-heuristics a suitable one to apply to a new problem. In this paper, we examine this idea further, looking more closely at the key features involved in measuring the similarity of exam timetabling problems.

In the following section we examine the various features of exam timetabling problems in a qualitative manner with a view to developing a measure of similarity between problems based on their most important features for use within our CBR system. With a number of potentially very different techniques within the case-base, we aim to develop a case-based reasoning system which will allow the user to input their needs and which can, (if the user desires it) return a set of complete timetables of similar quality from which the human timetabler can select the best one. If none of the timetables produced are acceptable, the user can change the focus of the constraints and ask the system to produce a new set of solutions from which to choose. By learning from its past experiences, the system will select from its case-base of heuristic and meta-heuristic techniques the one which best fits the new problem. Feedback from the human user as to the quality of the timetable(s) produced is also important for the system to learn whether the match was a good one or whether re-indexing is required to avoid the chosen heuristic being selected again for the given problem.

## 2 Studying the features of Exam Timetabling problems

### 2.1 The requirements of a CBR system

As mentioned earlier, our aim is to produce a case-based reasoning (CBR) heuristic/meta-heuristic selector which will intelligently choose, from a variety of techniques, the one best suited to a new problem given to the system. This is done by a matching process which employs a similarity measure between exam timetabling problems. Within the case base, each case is made up of a set of feature-value pairs representing a given problem, together with the heuristic or meta-heuristic technique(s) which give the best results for that problem. When presented with a new problem, its set of feature-value pairs is matched with those of all cases within the case-base and the most similar case(s) will be retrieved. The similarity measure employed will calculate a weighted sum of the difference in value for each feature between the two problems under comparison and is shown in equation 1:

$$S(C_x, C_y) = g(\sum_{i=1}^{n} h_i(|fx_i - fy_i|)) \quad (1)$$

where:

- $n$ is the number of features in the similarity measure
- $fx_i$ and $fy_i$ are the values of the $i$th feature of cases $C_x$ and $C_y$ respectively
- $h_i(a) := w_i a^2$; $w_i$ representing the weight of the $i$th feature
- $g(b) := 1/\sqrt{b+1}$

The similarity, $S(C_x, C_y)$, between two cases will be in the interval $(0, 1]$, with 1 representing two identical cases and results closer to 0 indicating cases with a low degree of similarity.

Clearly the key elements of this similarity function are the features $fx_i, fy_i$ of the two problems being compared together with their weights representing the importance of each feature to the overall similarity measure. These features will each be represented by a numerical value which can easily be compared to give a difference in value between the same feature (e.g. number of exams) across the two problems. To decide exactly which features should be used in the case-base and how to weight them is a near impossible task to perform purely by hand. Instead, we use knowledge discovery techniques in two stages, as reported by Burke et al. [17]. An initially large set of features and cases is systematically trimmed leaving only those cases which contribute positively to the knowledge of the system and identifying which combinations of features give the best system performance. In [17], the system performance is measured as being the number of successful retrievals of one of the best two heuristics as pre-calculated for a set of training cases and a set of test cases. Using these knowledge discovery techniques the system can tune the weightings on the features used as well as selecting the best subset of features themselves to improve the performance of the case base. Tabu Search and Hill Climbing techniques are both used within the knowledge discovery to obtain the best feature vector from the search space of all possible feature vectors. It is found that the best system performance comes from having a relatively small number of features within the case-base (usually between 3 and 7). Fewer features give too little information from which to accurately measure similarity, whilst too many features reduce system performance by diluting the impact of the most important features. In the following subsection we examine the initial list of simple features with which we begin the knowledge discovery process. Ratios between all pairs of simple features are also included within the features search space as many of these will provide far more meaningful measures of similarity than single features.

### 2.2 Qualitative analysis of features used within the CBR system

Whilst Burke et al. [17] concluded that between 3 and 7 features gives best performance for their CBR system, we initially wish to identify as many features as possible to use in the knowledge discovery process. In this way, a more thorough search can be performed to find the best feature vector to be used for our similarity measure given in equation 1. An analysis of the features used is given below, including combinations of features which may prove important.

***Number of Students*** By itself, the number of students within an exam timetabling problem can be extremely misleading as a measure of problem difficulty and as such it needs to be carefully considered before being input as a feature to be considered. In [14], the authors examine the role of the 'students' in the definition of an exam timetabling problem. It is evident that for real life problems with tight room capacity constraints resulting in a capacity constraint on each period of the timetable, the number of students is a crucial factor. However, it was noted that for the problems considered in that paper (no capacity constraints), the number of students was irrelevant. In such cases, the only role of the 'students' is to define the conflict matrix via their enrolments.

Having set up the conflict matrix, an *exams* x *exams* size matrix in which pairs of clashing exams are noted by a 1 with all non-clashing pairs being denoted by 0, the students play no further part in the problem and their number is unimportant. Indeed, there are a huge number of student enrolment sets which could define exactly the same conflict matrix - of particular note are the set of 2-enrolment students in which every student has just two enrolments and contributes to exactly 1 edge on the conflict graph[3]. In this case the number of students would be equal to the number of edges in the conflict graph (with weights on edges counting as multiple edges). At the other end of the spectrum would be the minimum set of 'students' which would define the same conflict matrix. This set would be obtained by assigning the largest clique of exams in the conflict graph to a student and then continuing, assigning the remaining largest clique to a new student until all exams and edges are assigned to a student. The relevance of this as a measure for similarity is considered later under the heading of "Cliques".

In those problems where capacity constraints are important, the number of students becomes more important, but only as a ratio to the total capacity available over the period of the timetable and in other ratios concerning enrolments for individual exams. Purely by itself, the number of students in a problem does not represent a feature which can contribute to a similarity measure.

***Number of Events (Exams)*** This is usually reported together with the number of students to give an idea of the size of a given data set. The exams form the core of the problem, being far more influential than the student numbers in defining the problem structure. The exam timetabling problem is concerned, of course, with assigning these exams to timeslots within a timetable with the

[3] The conflict graph is made up of vertices representing the exams in the problem, with an edge between any two exams which have students in common - the weight on the edge represents the number of students enroled to both exams

main constraint being that two clashing exams are not scheduled in the same slot. In a graph colouring model of exam timetabling (see [5]), the exams form the nodes of the graph with the edges, defined by the student enrolments, representing the conflicts. It is the structure of this graph representation which is one of the key aspects in how well a given heuristic or meta-heuristic technique will perform when addressing the problem - the number of nodes in the graph is one of the key aspects of this structure. The other main aspect is how these nodes are joined together by edges with some areas forming dense cliques and other areas being relatively sparse in terms of number of edges. These aspects will be considered further in the following sections of this paper. As a similarity feature by itself, results from the literature suggest that 'no. of exams' is a simple, yet effective indicator as to the potential best technique. The Great Deluge technique with adaptive initial solution generation presented by Burke and Newall [12] provides best known results at the time of writing on the 3 largest problems[4] from the Carter benchmark data sets [21] (see tables 1 and 2), each with $> 500$ exams. On some of the smaller problems, however, this technique proves less effective when compared to those of Caramia [18], Merlot et al. [29] and Casey and Thompson [24] (see table 3).

One major area which can be affected by the number of exams in the problem is the run time of a particular technique. In a majority of cases, more exams in the problem leads to longer running time of the algorithm or fewer iterations than would be possible in the same time on a smaller problem. As such, techniques which can converge to a good result in fewer iterations will have an advantage on larger problems. Other noteworthy features based on the number of exams include exams per student and exams per period averages. In the case of the Carter benchmarks, the STA-F-83 data set has a very large number of exams per student and exhibits very different behaviour from other problems. The GRASP technique of Casey & Thompson [24] gives a best known solution to this problem which is notably better than results reported from most other techniques. Clearly there are other factors which make this problem fairly anomalous amongst the 11 data sets considered however, as EAR-F-83 and YOR-F-83 also have a relatively large exam/student ratio whilst not exhibiting the same behaviour when addressed.

***Number of Periods*** The number of periods in a timetable can be either fixed *a priori* or it may be a variable to be minimised (as part of the objective function or separately). In this paper, we consider the case where the number of periods is fixed. For those problems in which the number of periods is not fixed, similarity can only be measured against other problems in which this is also the case. As a feature by itself, the number of periods assigned to a problem tells us next to nothing since it is only relevant in conjunction with other features to determine how highly constrained the problem is. In particular, the average number of exams per period and the ratio of number of clashes to number of periods may be of importance. Certainly, the number of exams per period gives a simple

[4] measured by number of exams

**Table 1.** Simple features for Carter Data Sets

| Data Set | No. of exams | No. of students | No. of enrolments | Graph Density | No. of periods |
|---|---|---|---|---|---|
| CAR-S-91 | 682 | 16925 | 56877 | 0.13 | 35 |
| CAR-F-92 | 543 | 18419 | 55522 | 0.14 | 32 |
| EAR-F-83 | 190 | 1125 | 8109 | 0.27 | 24 |
| HEC-S-92 | 81 | 2823 | 10632 | 0.42 | 18 |
| KFU-S-93 | 486 | 5349 | 25113 | 0.06 | 20 |
| LSE-F-91 | 381 | 2726 | 10918 | 0.06 | 18 |
| STA-F-83 | 139 | 611 | 5751 | 0.14 | 13 |
| TRE-S-92 | 261 | 4360 | 14901 | 0.18 | 23 |
| UTA-S-92 | 622 | 21267 | 58979 | 0.13 | 35 |
| UTE-S-92 | 184 | 2750 | 11793 | 0.08 | 10 |
| YOR-F-83 | 181 | 941 | 6034 | 0.29 | 21 |

**Table 2.** Further features for Carter Data Sets

| Data Set | enrolments per student | | Exams per period | enrolments per exam | | Largest no. of clashes for 1 exam |
|---|---|---|---|---|---|---|
| | average | max (num) | | average | max | |
| CAR-S-91 | 3.36 | 9 (1) | 19.5 | 83.40 | 1385 | 472 |
| CAR-F-92 | 3.01 | 7 (29) | 17.0 | 102.25 | 1566 | 381 |
| EAR-F-83 | 7.21 | 10 (9) | 7.5 | 44.80 | 232 | 134 |
| HEC-S-92 | 3.77 | 7 (1) | 4.5 | 131.26 | 634 | 62 |
| KFU-S-93 | 4.70 | 8 (11) | 24.3 | 51.67 | 1280 | 247 |
| LSE-F-91 | 4.01 | 8 (3) | 21.2 | 28.66 | 382 | 134 |
| STA-F-83 | 9.41 | 11 (209) | 10.7 | 41.37 | 237 | 61 |
| TRE-S-92 | 3.42 | 6 (20) | 11.4 | 57.09 | 407 | 145 |
| UTA-S-92 | 2.77 | 7 (23) | 17.8 | 94.82 | 1314 | 303 |
| UTE-S-92 | 4.29 | 6 (20) | 18.4 | 64.09 | 482 | 58 |
| YOR-F-83 | 6.41 | 14 (1) | 9.0 | 31.76 | 175 | 117 |

**Table 3.** Results from the literature (best results given)

| Data Set | Carter et al. | Caramia et al. | Burke & Newall | Di Gaspero | Casey & Thompson | Merlot et al. |
|---|---|---|---|---|---|---|
| CAR-S-91 | 7.1 | 6.6 | 4.6 | 5.7 | 5.4 | 5.1 |
| CAR-F-92 | 6.2 | 6.0 | 4.0 | - | 4.4 | 4.3 |
| EAR-F-83 | 36.4 | 29.3 | 36.1 | 39.4 | 34.8 | 35.1 |
| HEC-S-92 | 10.6 | 9.2 | 11.3 | 10.9 | 10.8 | 10.6 |
| KFU-S-93 | 14.0 | 13.8 | 13.7 | - | 14.1 | 13.5 |
| LSE-F-91 | 10.5 | 9.6 | 10.6 | 12.6 | 14.7 | 10.5 |
| STA-F-83 | 161.5 | 158.2 | 168.3 | 157.4 | 134.9 | 157.3 |
| TRE-S-92 | 9.6 | 9.4 | 8.2 | - | 8.7 | 8.4 |
| UTA-S-92 | 3.5 | 3.5 | 3.2 | 4.1 | - | 3.5 |
| UTE-S-92 | 25.8 | 24.4 | 25.5 | - | 25.4 | 25.1 |
| YOR-F-83 | 36.4 | 36.2 | 36.8 | 39.7 | 37.5 | 37.4 |

measure of one aspect of the *difficulty* of the problem and when combined with the conflict matrix density[5] this measure is potentially a very important one. The number of periods used for the benchmark data sets is slightly higher in each case than the minimum number of periods found to schedule all the exams in. However, the constraints are still very high since the objective function used for these problems is concerned with spreading clashing exams as far apart as possible in the timetable. Clearly, the more periods available in the timetable over the minimum required to obtain a feasible solution, the more the exams can be spread out. As can be seen from table 2, the ratio of exams to periods varies hugely between problems. The main reason for this is the difference in the conflict graphs for the problems. More highly conflicting problems will result in lower exams per period since the large number of conflicts often makes it hard to schedule all the exams in fewer periods. One of the key strengths of the knowledge discovery techniques we use to select which are the important features is that they can combine pairs of simple features which we select and chose amongst the resulting large number of features the most promising feature vector. This may include combinations of two simple features which we would not have otherwise considered. In problems where the number of rooms forms a tight constraint on the problem, the ratio between exams per period and the number of available rooms per period will also form an important characteristic of the problem.

***Conflict Matrix Density*** The conflict matrix, as defined earlier is an *exams* x *exams* size matrix in which pairs of clashing exams are noted by a 1 with all non-clashing pairs being denoted by 0. The conflict matrix density gives the ratio of 1's as a fraction of the total matrix. Therefore, a high conflict matrix density represents a high probability of conflict between any two exams. For example, a density of 0.5 implies that on average each exam will clash with half of the other exams in the problem. From tables 1 and 2, it can be seen that there is a strong correlation between exams per period and conflict matrix density. As mentioned above, this is due to the fact that a densely conflicting exam graph tends to mean that a higher number of periods are needed than sparsely conflicting graphs of the same size and therefore the average number of exams per period is correspondingly lower.

The conflict matrix is one of the most important aspects of any exam timetabling problem, representing both the hard constraints and some major soft constraints. Problems with a very high conflict matrix density, such as HEC-S-92 tend to be more difficult to find feasible solutions to in the first place. Also, for meta-heuristics which work only in the space of feasible solutions, this can lead to the search space being very disconnected with respect to certain move neighbourhoods. This can have a major impact on how successfully a particular meta-heuristic can traverse the search space to find a high quality solution. Also, the more disconnected the search space is for a given local search technique, the more focus is placed on the initial solution that is fed into the local search. While

[5] discussed in more detail later on

some techniques are relatively independent of initial solution and can connect the majority of the search space very effectively, others rely heavily on a good initialisation.

Conflict matrix density is one of the simplest metrics to be taken from the conflict matrix and also one of the most effective in measuring problem *difficulty*. However, it only gives an average on the percentage of other exams that each exam will clash with. Two problems with the same conflict matrix density can still be very different in structure. We discuss below some of the other measures, which when combined with conflict matrix density, should give a better indication of problem structure for the purposes of measuring similarity.

***Largest Degree*** The *degree* of an exam is defined as the number of other exams in the problem with which it conflicts through having students in common. One of the most common graph heuristics for constructing initial solutions for local search techniques uses a largest degree ordering of the exams to assign sequentially to the timetable. In this way, the most conflicting exams are assigned first as these are deemed to be the most difficult to schedule. The largest degree of an exam timetabling problem is the largest degree that occurs in any of the exams in the problem. As another measure to be obtained from the conflict matrix, this gives us some more information on the problem structure, but by itself of course this is not enough to be of use. The number of exams of largest degree could provide useful information, but is in most cases equal to one.

Statistical measures resulting from the conflict matrix are of more interest to us as features of the timetabling problem. The density mentioned earlier provides an average degree, with the largest degree giving us a maximum. In order to distinguish between two very different problems of equal conflict matrix density we need to consider statistical measures such as the variation in degree from the average and the degree of the exam at different percentiles[6]. We could also consider, as a variation, the number of exams or the percentage of the total exams whose degree is within a given percentage of the largest degree. As with all areas of statistical analysis, there is a large number of different statistical measures we could take based on the structure of the conflict matrix which could each provide some useful information relevant to measuring the similarity between two problems. For our work, we will concentrate on just a small number of these to be fed into the knowledge discovery process.

All of the above statistical measures can, of course, be combined as a ratio with our other features and of particular note would be those ratios to the total number of periods in the timetable. Also, a relatively simple measure which could provide useful knowledge is the percentage of the total exams whose degree is strictly less than the number of periods. Depending on the neighbourhood used, exams which clash with another exam in every period of the timetable may be unable to move within the local search process if the search is conducted within the set of feasible solutions only. For instance, many techniques successfully utilise the most simple move neighbourhood which selects a single exam and

[6] with exams ordered in decreasing order of degree

moves it to a new period of the timetable, selected either at random or by some deterministic method. Using this neighbourhood, only those exams which do not have a clash in every other period of the timetable can be moved which can lead to a large amount of disconnectivity in the search space. On the other hand, using a neighbourhood such as the Kempe Chain neighbourhood employed by Thompson & Dowsland [39] and Casey & Thompson [24] ensures that every exam within the timetable can move to any other timeslot. In doing so, a series of other exams will also often have two be exchanged between the two periods in question. This is investigated further in the following section.

***Fluidity Analysis*** In table 4, we present an analysis of the fluidity of the benchmark problems studied when optimised using Simulated Annealing with the simplest move neighbourhood[7] over 100 runs, each with a different initial solution. The initial temperature and cooling schedule were set to be extremely high and slow respectively for these experiments since our aim was to examine how many exams within the timetable never moved from their initial position as given by our largest degree construction heuristic. With such a high temperature and slow cooling schedule, we can say with a relatively high degree of certainty that any exam which is capable of moving within the neighbourhood used would do so at some point over the course of the 100 separate runs. Of course, there will be exams which may have a small window of opportunity to move in this neighbourhood, when a period briefly becomes available that they can move to, but given the random nature of the move selection, the exam was not selected to be moved during this window. However, such exams will be very few across 100 runs of the algorithm. Our main objective was to examine how different the data sets are with respect to fluidity for this commonly used neighbourhood.

**Table 4.** Percentage of total number of exams which never move in $x$ runs out of 100 of Simulated Annealing using the simple move neighbourhood

| Data | No. of runs in which $x$% of exams never moved | | | | | |
|---|---|---|---|---|---|---|
| Set | 100 | 75-99 | 50-74 | 25-49 | 1-24 | 0 |
| CAR-S-91 | 1.17% | 3.96% | 3.96% | 3.81% | 10.12% | 76.98% |
| CAR-F-92 | 3.31% | 3.31% | 2.76% | 3.50% | 9.39% | 77.72% |
| EAR-F-83 | 0.55% | 6.63% | 0.55% | 2.21% | 10.50% | 79.56% |
| HEC-S-92 | 0.00% | 1.23% | 3.70% | 2.47% | **50.62%** | 41.98% |
| KFU-S-93 | 1.65% | 2.88% | 1.44% | 3.70% | 4.53% | 85.80% |
| LSE-F-91 | 1.57% | 4.99% | 1.57% | 1.31% | 3.15% | 87.40% |
| STA-F-83 | **37.41%** | 0.00% | 0.00% | 3.60% | 1.44% | 57.55% |
| TRE-S-92 | 0.77% | 2.68% | 0.38% | 1.15% | 7.66% | 87.36% |
| UTA-S-92 | 3.05% | 5.31% | 3.05% | 1.93% | 7.07% | 79.58% |
| UTE-S-92 | 6.52% | 0.54% | 1.63% | 1.09% | 5.43% | 84.78% |
| YOR-F-83 | 0.53% | 1.58% | 0.53% | 0.00% | 8.42% | 88.95% |

[7] Move a single exam to a new timeslot whilst maintaining solution feasibility

It is clear that in the case of exams which never move throughout the local search process, their positioning in the initial solution is crucial to the quality of the final solution. For the majority of data sets in table 4, this percentage of exams is relatively small and generally below $\sim 3\%$. There are also a relatively small percentage of exams which fail to move in $> 75$ of the 100 runs. A higher percentage are immobile in $< 25$ runs, but in most cases $\sim 80\%$ of the exams in the data sets move at least once in every one of the 100 runs. Of course, our analysis does not tell us whether many of the exams moved just once during the local search or whether they moved hundreds of times, but in this analysis we are mostly interested in the boolean variable of whether an exam moved at all or not. Whilst 9 out of the 11 benchmark data sets presented share fairly similar fluidity analysis which does not add much to our similarity measurement, 2 of the data sets exhibit very different behaviour.

Standing out most obviously are the $\sim 37\%$ of exams in the STA-F-83 data set which never move across any of the 100 runs. Also notable is the fact that there are 0 exams in the 50-99 group and very few in the 1-49 group. This indicates that if an exam can move at all in the STA-F-83 data set within this simple move neighbourhood, it will tend to do so in the vast majority of runs. Having over one third of its exams immobile relative to this simple move neighbourhood provides some important clues as to the anomalous nature often displayed by this data set. The reliance on the initial solution becomes massive with so many exams being set in the positions that they are originally placed in. Coupled with the fact that each student takes an average of 9-10 exams and these are spread across just 13 time slots, it is easy to see why this data set yields a very high penalty cost for all feasible solutions. Those exams which are immobile will, in general, be the ones with the most clashes and which therefore add the most penalty to the timetable. With this in mind, one of the reasons we believe Casey & Thompson's GRASP technique [24] is so successful on this problem relative to other techniques is the implementation of the Kempe Chain based neighbourhoods. As discussed earlier, the Kempe Chain neighbourhood allows any exam in the timetable to be moved to any other timeslot whilst always yielding a feasible solution. Using the simple neighbourhood, if the exam, $e$ in timeslot $t_1$to be moved clashes with an exam in the chosen slot, $t_2$, it cannot be moved there and a new move must be selected. Kempe chains get around this problem by moving all those clashing exams from $t_2$ across to $t_1$. Any further clashes induced by this are resolved by moving the clashing exams across with the original exam $e$ to timeslot $t_2$, with this process continuing until the two periods are conflict-free. Due to the fact that all periods are conflict-free before the first exam is moved, there will always exist a feasible resolution to the Kempe Chains. In the worst case scenario this would involve swapping all exams in $t_1$ with all exams in $t_2$.

The other data set of note is HEC-S-92 whose behaviour is perhaps even more interesting than that of STA-F-83 and also less easily understandable. Contrary to the other 10 data sets, HEC-S-92 does not have a single exam which never moves across 100 runs of Simulated Annealing from a different initialisation each time. The percentage of exams which fail to move in 25-99 runs is also very low, yet over half the exams (41) in the data set fail to move in 1-25 of the 100 runs. A deeper analysis of this behaviour concerning how much overlap there is in the runs during which these $> 50\%$ of exams do not move would be required to draw any firm conclusions about this behaviour, but it is worthy of note as being significantly different from all other data sets. This data set is also the only one which our Greedy Largest Degree with Backtracking initialisation technique (used to provide initial solutions for local search techniques for use within our CBR system [14–16]) fails to find a feasible initial solution to. This is due to the fact that our backtracking only searches 2 levels deep before giving up and restarting, but on this data set it always reaches the same irresolvable point. This is probably due to a combination of the high conflict matrix density and also the unusual behaviour indicated by our fluidity analysis. What this fluidity analysis seems to show is that whilst all exams in the data set can move around over the course of 100 runs from random initialisations, the actual fluidity of the data set from any given initialisation is not so high with a certain number of exams being fixed by their relative positions to other exams. Again, this behaviour can be attributed to the fact that the conflict graph is very dense. An analysis of cliques within the problem as discussed in the following subsection may also shed some light on this behaviour.

From this analysis it would seem that the fluidity of a given data set with respect to the neighbourhood used in a meta-heuristic can prove crucial to how successful the meta-heuristic will be. From the point of view of our similarity measure, this is an area which could prove very important with further analysis to be carried out.

***Cliques*** One of the most significant features of exam timetabling problems when modeled as a graph is that large cliques and near-cliques tend to exist, unlike the structure of a typical random graph in which any two nodes have an equal probability of being connected by an edge. In exam timetabling, many of the edges which contribute to the conflict matrix are clustered together representing exams which form part of a particular discipline. Students taking science-based subjects will generally take very few, if any, humanities exams, but will take a large number of science exams meaning that the density of clashes between science exams will be far greater than between science exams and humanities exams. Carter & Johnson [23] investigate the cliques to be found within the benchmark problems considered in this paper as well as looking at near-maximum cliques. An investigation of cliques gives another angle to measuring similarity (based upon the conflict matrix). As well as considering the size of the maximum clique in each problem graph, Carter & Johnson investigate how many cliques there are of max size and also (max-1) size. As a problem feature, the number of maximum size cliques could provide invaluable information for measuring similarity between problems by giving a much more in-depth view of the structure of the conflict matrix.

The two largest data sets (CAR-S-91 and UTA-S-92) are found to have over 100 maximum size cliques whilst the majority of data sets have fewer than 5. Again the STA-F-83 data set exhibits very different behaviour to the other data sets, being the second smallest measured by exam size, but having 60 cliques of size 13, which is also the number of periods used to construct the timetable. Carter & Johnson also calculate the number of nodes occurring in all max cliques and in any max clique together with an analysis of the complement graphs. When considering cliques of size (max-1), the number of these is significantly larger than max-size cliques in the majority of problems. The authors move on to consider Quasi-cliques, where all nodes in a quasi-clique, $Q_k$, have at most $k$ missing edges from a true clique. Again these represent very dense areas of the conflict graph which have a far larger impact on the difficulty of the problem than the much less dense areas which balance these out to give the overall conflict matrix density.

There is a large amount of analysis which can potentially be carried out on cliques, with Carter & Johnson's work [23] providing a crucial backbone for this. How much of the clique analysis could be used as part of our similarity measure remains to be seen, however, since finding the largest clique in a graph of size $n$ is in itself an NP-hard problem. As such, it may not be feasible for our CBR system to calculate the required feature data for a new problem in order to compare with those in the case base. Having said that, cliques clearly form the basis of the core problem definition in most exam timetabling problems so cannot be ignored.

***Side constraints & the Objective function*** So far we have considered features which are common to the core exam timetabling problem where the only hard constraints are that every exam must be scheduled to exactly one timeslot within the timetable and no two exams with students in common can be scheduled in the same time slot. The only soft constraint so far considered is that of spreading clashing exams around the timetable using the proximity cost given by Carter et. al [21]. In reality, real world problems will have a number of other constraints, both hard and soft. The hard constraints will determine the feasible solutions space, whilst the soft constraints will give a measure of how good the timetable is, either by being combined in a weighted single objective function or by forming a pareto front in a multi-objective optimisation (e.g. [3, 37]).

When attempting to measure similarity between exam timetabling problems, it is required that the problems being compared have the same constraints. Attempting to compare two problems, one in which the number of periods in the timetable is fixed and one in which it is a variable to be minimised is clearly not sensible. For this reason, we need within the case-base a large variety of problems with different hard and soft constraints to give the system as wide applicability as possible. This can be done potentially by adding in constraints to the core problems and forming new cases with these additional constraints.

The authors' previous work investigating the effect of the objective function on potential similarity measures [15] showed that, even when the same soft constraints are employed within the problem, using a different set of weights on the constraints can have an effect on the performance of a given heuristic applied to the problem. This is to be expected since the objective function defines the height of the problem landscape at every point, therefore using different weights on the same set of constraints could change the structure of the landscape significantly causing a meta-heuristic which may previously have traversed the landscape very effectively to now get stuck more in local optima and provide a less high quality result.

From this analysis, it would seem that the matching process employed within our CBR system can only compare two problems whose definitions are the same regarding the hard and soft constraints included as well as the weightings applied to those soft constraints. It may be possible to adapt problems within the case base to use the weightings of a new problem given to the system with the same constraints, but this would require more research.

## 3 Conclusions and future work

In this paper we presented a discussion of the key features of exam timetabling problems. The motivation for this is the aim of creating a similarity measure between timetabling problems, which can be used within a case-based reasoning (CBR) system to intelligently select a heuristic or meta-heuristic technique to solve a new problem. We selected a number of simple features which we expect to form a major part of this similarity measure and analysed their contribution and importance to the problem solving method. From the outset we knew that the contribution to a similarity measure of the simple features listed would by itself be quite small, but that the contribution of combinations of these features would form the key to the similarity measure.

Of the simple features themselves, the number of exams is considered to be a very good basic indicator of problem difficulty and also of which techniques are likely to be successful. In particular, it was noted that on the larger problems (500+ exams), the hybrid great deluge meta-heuristic [12] always outperforms the other techniques considered. For problems of smaller size, other features become more important to distinguish between different problems. The number of students in the problem and the number of periods in the problem are not considered to be of any value as features by themselves, but combined in ratios with other features, number of periods is an important factor. Of our other features studied, the conflict matrix density is thought to be an important element of a problem definition, providing a basic measurement of how highly constrained the problem is. However, we also noted that other features of the conflict matrix are also required to get a better indication of the problem structure. Amongst these are the variation in degree of exams from the average and the cliques found within the conflict matrix.

A knowledge discovery process will be applied (similar to [17]), which will select from a large number of features (including ratios of all pairs of features included in the system), those which provide the best measure of similarity between exam timetabling problems. It has been shown [17] that between 3 and 7 features tend to give the best performance for a similarity measure and it is thought likely that most of these features will be ratios of the more simple features presented in this paper, some of which may not have been considered to be important, but which may be found to provide crucial knowledge of the problems studied. Ultimately however, it will be the training within the knowledge discovery process which will conclude which combinations of features give best system performance based on our test cases.

Future work on the similarity measure may include investigating the use of Fuzzy Sets for indexing and retrieval of cases within CBR [28]. This technique has numerous advantages, allowing numerical features to be simplified into a number of fuzzy sets for comparison as well as making it possible to index a case multiple times on a given feature with different degrees of membership. The use of fuzzy sets would allow more flexibility in the comparison of features with matches being suggested which might not otherwise have come up as the most similar case.

## References


1. H. Asmuni, E. K. Burke & J. M. Garibaldi: Fuzzy Multiple Ordering Criteria for Examination Timetabling. To appear in the Proceedings of the 5th international conference on the Practice and Theory of Automated Timetabling (PATAT), Pittsburgh, USA, August 18th-20th, 2004.
2. V. A. Bardadym: Computer-Aided School and University Timetabling: The New Wave. In [30], pages 22-45, 1996.
3. E.K.Burke, Y. Bykov & S.Petrovic: A Multicriteria Approach to Examination Timetabling. In [32], pages 104-117, 2001.
4. E. K. Burke, D. G. Elliman, P. H. Ford & R. F. Weare: Examination Timetabling in British Universities: a Survey. In [30], pages 76-90, 1996.
5. E. K. Burke, J. H. Kingston and D. De Werra: Chapter 5.6: Applications to Timetabling, In the Handbook of Graph Theory, (eds. J. Gross and J. Yellen), pp 445-474, Chapman Hall/CRC Press, 2004.
6. E. K. Burke, J. H. Kingston, K. Jackson & R. Weare: Automated University Timetabling: The State of the Art. The Computer Journal 40 (9), pages 565-571, 1997.
7. E. K. Burke, J. P. Newall & R. F. Weare: A Memetic Algorithm for University Exam Timetabling. In [30], pages 241-250, 1996.
8. E. K. Burke, J. P. Newall & R. F. Weare: Initialization Strategies and Diversity in Evolutionary Timetabling. Evolutionary Computation 6(1), pages 81-103, 1998.
9. E. K. Burke & J. P. Newall: A Multi-stage Evolutionary Algorithm for the Timetable Problem. IEEE Transactions on Evolutionary Computation, 3(1), pages 63-74, 1999
10. E. K. Burke, B. MacCarthy, S. Petrovic & R. Qu: Structured Cases in Case-Based Reasoning - Re-using and Adapting Cases for Time-tabling Problems. Knowledge-Based Systems, 13: 2-3 (2000) pages 159-165. Also in proceedings of ES'99 as one of the best papers.
11. E. K. Burke, B. MacCarthy, S. Petrovic & R. Qu: Case-based Reasoning in Course Timetabling: An Attribute Graph Approach. Case-Based Reasoning Research and

Development, Proceedings of the 4th International Conference on Case-Based Reasoning (ICCBR-2001). Vancouver, Jul-Aug, Canada, 2001. LNAI 2080. pages 90-104
12. E. K. Burke & J. P. Newall: Enhancing Timetable Solutions with Local Search Methods. In [33], pages 195-206, 2003
13. E. K. Burke, S. Petrovic & R. Qu: Case-Based Heuristic Selection for Examination Timetabling. Proceedings of the SEAL'02 conference, Singapore, pages 277-281, 2002.
14. E. K. Burke, A. J. Eckersley, B. McCollum, S. Petrovic & R. Qu: Similarity Measures for Exam Timetabling Problems. 1st Multidisciplinary Intl. Conf. on Scheduling: Theory and Applications (MISTA 2003), Vol. 1, pages 120-136, ISBN0-9545821-0-1, Nottingham, UK, Aug 13-16, 2003.
15. E. K. Burke, A. J. Eckersley, B. McCollum, S. Petrovic & R. Qu: Using Simulated Annealing to Study Behaviour of Various Exam Timetabling Data Sets. Proceedings of the Fifth Metaheuristics International Conference (MIC 2003), paper MIC03 09, Kyoto Japan, Aug 2003.
16. E. K. Burke, A. J. Eckersley, S. Petrovic & R. Qu: Investigating Variable Neighbourhood Search for exam timetabling. University of Nottingham Working Paper NOTTCS-WP-2004-1, in preperation.
17. E. K. Burke, S. Petrovic & R. Qu: Case-based Heuristic Selection for Timetabling Problems. University of Nottingham Technical Report NOTTCS-TR-2004-2, accepted for publication in the Journal of Scheduling, 2005.
18. M. Caramia, P. Dell'Olmo & G. F. Italiano: New Algorithms for Examination Timetabling. In S. Näher & D. Wagner (eds.): Algorithm Engineering 4th Int. Workshop, Proc. WAE 2000 (Saarbrücken, Germany). Lecture Notes in Computer Science, Vol 1982. Springer-Verlag, Berlin Heidelberg New York. pages 230-241, 2001.
19. M. W. Carter: A Survey of Practical Applications of Examination Timetabling Algorithms. In Operations Research 34, pages 193-202, 1986.
20. M. W. Carter & G. Laporte: Recent Developments in Practical Examination Timetabling. In [30], pages 373-383, 1996.
21. M. W. Carter, G. Laporte & J. W. Chinneck: A General Examination Scheduling System. Interfaces 24, pages 109-120, 1994.
22. M. W. Carter, G. Laporte & S. Y. Lee: Examination Timetabling: Algorithmic Strategies and Applications. Journal of the Operational Research Society 47, No. 3, pages 373-383, 1996.
23. M. W. Carter & D. G. Johnson: Extended clique initialisation in examination timetabling. Journal of the Operational Research Society (2001) 52, pages 538-544.
24. S. Casey & J. Thompson: GRASPing the Examination Scheduling Problem. In [33], pages 232-244, 2003.
25. T. B. Cooper & J. H. Kingston: The complexity of timetable construction problems. In [30], pages 283-295, 1996.
26. D. de Werra: An Introduction to Timetabling. European Journal of Operational Research, 19, pages 151-162, 1985.
27. E. Foxley & K. Lockyer: The Construction of Examination Timetables by Computer. The Computer Journal 11, pages 264-268, 1968.
28. B. C. Jeng & T-P. Liang: Fuzzy Indexing and Retrieval in Case-Based Systems. Expert Systems With Applications, Vol 8, No. 1, pages 135-142, 1995.
29. L. T. G. Merlot, N. Boland, B. D. Hughes & P. J. Stuckey: A Hybrid Algorithm for the Examination Timetabling Problem. In [33], pages 207-231, 2003.

30. E. K. Burke & P. Ross (eds): Practice and Theory of Automated Timetabling. Volume 1153 of Lecture Notes in Computer Science. Springer-Verlag, Berlin, Heidelberg, 1996.
31. E. K. Burke & M. W. Carter (eds): Practice and Theory of Automated Timetabling. Volume 1408 of Lecture Notes in Computer Science. Springer-Verlag, Berlin, Heidelberg, 1998.
32. E. K. Burke & W. Erben (eds): Practice and Theory of Automated Timetabling. Volume 2079 of Lecture Notes in Computer Science. Springer-Verlag, Berlin, Heidelberg, 2001.
33. E. K. Burke & P. De Causmaecker (eds): Practice and Theory of Automated Timetabling. Volume 2740 of Lecture Notes in Computer Science. Springer-Verlag, Berlin, Heidelberg, 2003.
34. L. Di Gaspero: Recolour, Shake and Kick: A recipe for the Examination Timetabling Problem. In: E. K. Burke & P. De Causmaecker (eds.): Proceedings of the Fourth International Conference on the Practice and Theory of Automated Timetabling, Gent, Belgium, August 2002, pages 404–407.
35. A. Schaerf: A Survey of Automated Timetabling. In Artificial Intelligence Review, 13(2), pages 87-127, 1999.
36. L. Di Gaspero & A. Schaerf: Tabu Search Techniques for Examination Timetabling. In [32], pages 104-117, 2001.
37. S. Petrovic & Y. Bykov: A Multiobjective Optimisation Technique for Exam Timetabling Based on Trajectories. In [33], pages 181-194, 2003.
38. S. Petrovic and R. Qu: Case-Based Reasoning as a Heuristic Selector in a Hyper-Heuristic for Course Timetabling Problems. Proceedings of the 6th International Conference on Knowledge-Based Intelligent Information Engineering Systems and Applied Technologies (KES'02), Vol. 82, Milan, Italy, pages 336-40, Sep 16-18, 2002.
39. J. Thompson & K. Dowsland: Variants of Simulated Annealing for the Examination Timetabling Problem. Ann. Oper. Res. 63, pages 105-128, 1996.
40. J. Thompson & K. Dowsland: A Robust Simulated Annealing Based Examination Timetabling System. Comput. Oper. Res. 25, pages 637-648, 1998.
41. G. M. White & B. S. Xie: Examination Timetables and Tabu Search with Longer-term Memory. In [32], pages 85-103, 2001.