A Practical Examination Timetabling Problem at the Universiti Kebangsaan Malaysia

Masri Ayob, Salwani Abdullah and Ariff Md Ab Malik

Faculty of Technology and Information Science, Universiti Kebangsaan Malaysia,
43600 UKM Bangi, Selangor, Malaysia

Summary
The examination timetabling problem represents a major administrative activity for academic institutions. An increasing number of student enrolments, a wider variety of courses and increasing number of combined degree courses contribute to the growing challenge of developing examination timetabling software to cater for the broad spectrum of constraints required by educational institutions. In this paper, we present a new real-world examination timetabling dataset at the University Kebangsaan Malaysia that will hopefully be used as a future benchmark problem. In addition, a new objective function that attempts to spread exams throughout the examination period is also introduced. This objective function that taking into account both timeslots and days assigned to each exam, is different from the often used objective function from the literature that only considers timeslot adjacency.

Key words:
Timetabling, Heuristic, Graph colouring

1. Introduction

Examination timetabling is concerned with an assignment of exams into a limited number of timeslots subject to a set of hard constraints (see Burke et al. [1]). Generally accepted hard constraints for the examination timetabling problem are: (i) no student should be required to sit two exams at the same time and (ii) the scheduled exams must not exceed the room capacity. However, in practical examination timetabling problem, there are many other constraints and the constraints vary among institutions. Similarly in our dataset, we have additional hard constraints (as described in section 2).

Hard constraints are rigidly enforced. Solutions satisfy all the hard constraints are called *feasible*. On the other hand, there might be some requirements that are not essential but should be satisfied as far as possible, which are referred to as soft constraints. A particularly common soft constraint refers to spreading exams as evenly as possible throughout the schedule. Due to the complexity of the problem, it is not usually possible to have solutions that do not violate all of the soft constraints. In fact, the cost function (that evaluates how good the solutions are) is a function of violated soft constraints. A weighted penalty value is associated with each violation of the soft constraint and the objective is to minimize the total penalty value.

A wide variety of approaches for constructing examinations timetables have been discussed in the literature. Carter [2] divided these approaches into four broad categories: sequential methods, cluster methods, constraint-based methods and meta-heuristics. Petrovic and Burke [3] added the following categories: multi-criteria approaches, case-based reasoning approaches and hyper-heuristics/self adaptive approaches.

In sequential methods, the construction of a conflict-free timetable is modeled as a graph coloring problem (see Burke et al. [4]). Clustering methods split exams into group (see Balakrishnan et al. [5], White and Chan [6]). David [7] and Boizumault et al. [8] applied constraint-based approaches to timetabling problems.

Meta-heuristic approaches (which include simulated annealing, tabu search, genetic algorithms and hybrid approaches such as memetic algorithms) have also been investigated in the last 20 years. Thompson and Dowsland [9] investigated a two phase simulated annealing approach. Examples of tabu search based approaches were presented by Di Gaspero and Schaerf [10] and White and Xie [11]. Hybridization techniques have been shown to perform well in examination timetabling (see Merlot et al. [12], Burke and Newall [13]).

Multi-criteria approaches to timetabling offer a more flexible way of handling different types of constraints simultaneously (see Petrovic and Bykov [14]). Case-based reasoning (see Burke et al. [15]) is an approach that is motivated by the human process of learning from previous experience and using that experience to solve new problems. Burke et al. [16] applied a case-based reasoning method to select examination timetabling heuristics. Fuzzy reasoning has recently been investigated with some success for examination timetabling by Asmuni et al. [17, 18].

Manuscript received Sept 5, 2007.
Manuscript revised Sept 20, 2007.

Hyper-heuristics are emerging as powerful approaches which raising the level of generality of timetabling systems (see Burke and Petrovic [19], Petrovic and Burke [3], Kendall and Hussin [20]). Burke and Newall [21] have presented an adaptive heuristic approach which draws upon the squeaky wheel optimization methodology developed by Joslin and Clements [22].

Interested readers can find more details about examination timetabling research in Schaerf [23], Carter [2], Carter and Laporte [24], de Werra [25], Bardadym [26], Burke et al. [27], Burke and Petrovic [28], Petrovic and Burke [3], Abdullah et al. [29, 30], Ayob et al. [31], Burke and Bykov [32], Burke et al. [33] and Asmuni et al. [34].

In this paper, we introduce a new real-world capacitated examination timetabling dataset at the University Kebangsaan Malaysia (UKM) that has more practical constraints (see section 2) compared to existing benchmark examination datasets. We hope that the dataset will be used as a future benchmark problem. The quality of the timetable is measured using a new objective function which is different from the standard proximity cost (as introduced by Carter et al. [35]), where the closeness of the scheduled exams is not only measured based on the assignment of the timeslots, but also on the assignment of the days. This new objective function can also be applied to the standard benchmark examination datasets (Carter et al. [35]) by adding a new variable *day* for each corresponding timeslot.

This paper is organised as follows: The next section presents the description of the problem. The formulation of the problem is outlined in Section 2. Section 4 introduces a new objective function followed by someconcluding remarks and new research directions in Section 5.

2. Problem Statement

In this paper, we study a real-world examination timetabling problem at the University Kebangsaan Malaysia (UKM). Particularly, the dataset (UKM06-1) presented here is real data for undergraduate examinations for Semester I, year 2006. The dataset presented here has been processed which excluded the courses with no exam and modified the original dataset by replacing the appropriate exams accordingly. In this dataset, the total number of examinations is 818 exams with 14047 students, 75857 enrollments and the number of exam days and timeslots are 15 days with 42 available timeslots. The dataset is available at http://www.ftsm.ukm.my/jabatan/tk/masri/Exam/.

In UKM, there are courses which have no exams. We also have co-curriculum courses (university level courses which are enrolled by many students from different faculties and have shorter exam periods, i.e. different timeslots), which have to be scheduled outside the examination weeks. These courses have to be excluded from our dataset. In UKM, we have two campuses, i.e. a main campus in Bangi and the KL campus. The dataset only considers the examinations in the Bangi campus (excluding examinations for a Law faculty because they have different timeslots). There are examinations which are equivalent and need to be scheduled together with other examinations. Before solving the problem, we need to replace the equivalent exam with one exam from each set.

In this problem, we have 3 week examination periods. Each week has 5 days (Monday to Friday). Each day has 3 timeslots, except Friday which has 2 timeslots. In order to closely model the real-world timeslots, we present the following vectors (Figure 1) which demonstrates the idea:

(1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4, 5, 5, 8, 8, 8, 9, 9, 9, 10, 10, 10, 11, 11, 11, 12, 12, 15, 15, 15, 16, 16, 16, 17, 17, 17, 18, 18, 18, 19, 19)

Fig.1. Vector of timeslots in day

It can be seen that there are only two “5” entries (two timeslots on the first Friday – the 5th day). Saturdays (days 6 and 13) and Sundays (day 7 and 14) are missing because there are no examinations on Saturday and Sunday. The corresponding timeslot vector is presented in Figure 2.

(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 36, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 57)

Fig. 2. Vector of timeslots

In Figure 2, the timeslots are represented as indexes. Timeslots 1, 2 and 3 are referring to day 1, timeslots 4, 5 and 6 are referring to day 2, etc. Note that on day 5 (Friday the first week) there are only 2 timeslots i.e. 13 and 15 (morning and evening sessions). Since there is no afternoon session, so we do not use a timeslot index 14. The reason for this representation is that we want to give a suitable weight according to the actual time gap since in this case, students actually have one free slot (i.e. 2 gaps in this case, 15-13=2). We also can see that the timeslot indexes for Saturday, the first week (16, 17 and 18), and Sunday, the first week (19, 20, 21), are missing because there are no exam scheduled on Saturday and Sunday. The same representation is used for the second and third weeks of exam period. The idea is to reflect the time slot gap with the practical time gap. That is, in the real situation, for example, we have 2 days (weekend) break between the exam on Friday evening (timeslot 15 and 36) and Monday morning (timeslot 22 and 43). Therefore, it is not appropriate to index the Friday evening and Monday morning time slot as adjacent slots if in real situation there is no exam on Saturday and Sunday. This indexing format (day vector and timeslot vector) could also be applied to other datasets, including the benchmark datasets by adding day vectors for each timeslot and introducing missing timeslot if there are free exam timeslots (e.g. weekend break or public holiday). The number of timeslots per day is institution dependent. Therefore, for example, if the institution has two timeslots per day, we should only have two day vectors for each day.

Room specifications are shown in table 1. Each examination should be assigned to a single room, unless this cannot be avoided. In exceptional cases, i.e. no room available to fit the exam, then the exam can be assigned to multiple rooms but the room location should be closed to each other, for example, in this case, it should be in DECTAR (starting with the largest room in DECTAR i.e. Dewan(DECTAR), LobiUtama (DECTAR), PSeni(DECTAR), LobiA(DECTAR) and LobiB(DECTAR)). The constraint is enforced due to the location practicality. However, for the case of large examinations, where the number of enrolments is greater than the largest room capacity (i.e. more than 850 seats in this case), the examination can be assigned to any available room starting with DPBestari, DGemilang, Dewan (DECTAR), LobiUtama (DECTAR), PSeni (DECTAR), LobiA (DECTAR) and finally LobiB (DECTAR)). The room can be shared with multiple exams depending on the availability of the seats. However, in assigning exams to rooms, priority should be given to assign an exam to a room which can accommodate the exam. In addition, wherever possible, students should be assigned to the same room when they are sitting consecutive exams on the same day. Apart from that, Law courses (which were excluded in this dataset, but need to be scheduled later by considering pre-assigned exams) need to be assigned to PSeni (DECTAR) only. For evening session, Law exams cannot share room with other courses because they have different starting time for evening session.

Table 1: Available rooms for dataset UKM06-1

Room	**Room capacity**
DPBestari	850
DGemilang	610
Dewan (DECTAR)	610
LobiUtama (DECTAR)	270
PSeni (DECTAR)	152
LobiA (DECTAR)	70
LobiB (DECTAR)	70

To facilitate future researchers, who might be interested in using UKM06-1 dataset, we present details of the file format of the dataset in the appendix.

3. Problem Formulation

The examination timetabling problem can be stated as follows:

- N is the number of exams;
- E_i is an exam where $i \in \{1,....,N\}$;
- e_i is number of students sitting exam E_i where $i \in \{1,....,N\}$;
- B is the set of all N exams, $B=\{E_1,..., E_N\}$;
- D is the number of days;
- T is the given number of available timeslot;
- M is the number of students;
- R is the number of available rooms;
- L_f is the capacity of room f where $f \in \{1,....,R\}$;
- r_i specifies the assigned room for exam E_i, where $r_i \in \{1,....,R\}$ *and* $i \in \{1,....,N\}$;
- t_i specifies the assigned time slot for exam E_i, where $t_i \in \{1,..,T\}$ and $i \in \{1,..,N\}$;
- d_i specifies the assigned day for exam E_i, where $d_i \in \{1,..,D\}$ and $i \in \{1,..,N\}$;
- $C=(c_{ij})_{NxN}$ is the conflict matrix where each element denoted by c_{ij}, $(i,j \in \{1,..,N\})$ is the number of students taking exams E_i and E_j;
- $\Delta t =|t_i-t_j|$ is the timeslot different between exam E_i and E_j;
- $\Delta d=|d_i-d_j|$ is the day different between exam E_i and E_j;
- z_i is a lecturer for exam/courses E_i.

The constraints of our dataset are:

1) All exams must be scheduled and each exam must be scheduled only once.

$$\sum_{s=1}^{T} \lambda_{is} = 1 \quad \text{for all } i \in \{1......N\} \tag{1}$$

where

$$\lambda_{is} = \begin{cases} 1 & \text{if exam } i \text{ is assigned to} \\ 0 & \text{otherwise;} \end{cases} \tag{2}$$

2) No student can sit in two exams concurrently. If examination i and j are scheduled in slot s, the number of students sitting both examination i and j must be equal to zero, i.e. $c_{ij}=0$.

$$\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} c_{ij}.x(t_i, t_j) = 0 \tag{3}$$

where

$$x(t_i, t_j) = \begin{cases} 1 & \text{if} \quad t_i = t_j; \\ 0 & \text{otherwise;} \end{cases} \tag{4}$$

3) For each timeslot t, the number of students sitting exams ($Students_t$) must not exceed the maximum seat number ($Seats$) i.e. 2400 seats per slot for this case study.

$$Students_t \le Seats \quad \text{for } t \in \{1....T\}; \tag{5}$$

4) Student which has consecutive exams on the same day should be assigned to the same room, i.e. both exams are assigned to the same room.

$$\text{if } t_i = x;\ t_j = x+1;\ d_i = d_j \text{ and } c_{ij} \neq 0 \text{ then } r_i = r_j \text{ for all } i, j \in \{1,....,N\}; \tag{6}$$

5) Special examination, $E_i \in S$ where $S \subset B$ should be isolated from other exams (e.g. in UKM06-1 dataset, exam VVVA3213 requires audio), i.e. the special exam cannot share room with other exam at the same timeslot.

$$\sum_{i=1}^{N} \alpha_{ir} \le 1 \quad \text{for all } r \in \{1......R\} \tag{7}$$

where

$$\alpha_{ir} = \begin{cases} 1 & \text{if exam } E_i \in S \text{ is assigned to room } r; \\ 0 & \text{otherwise;} \end{cases} \tag{8}$$

6) No student can seat 3 consecutive exams in a day.

$$\text{If } c_{ij} \neq 0;\ c_{ik} \neq 0;\ t_i = x;\ [t_j = x+1 \text{ OR } t_j = x-1] \text{ and } d_i = d_j \text{ then } d_k \neq d_i; \text{ for all } i,j \in \{1,...,N\}; \tag{9}$$

7) Wherever possible, each examination must be assigned to a single room.

$$\sum_{f=1}^{R} \beta_{if} = 1 \quad \text{for all } i \in \tag{10}$$

where

$$\beta_{if} = \begin{cases} 1 & \text{if exam } i \text{ is assigned to room } f; \\ 0 & \text{otherwise;} \end{cases} \tag{11}$$

8) Exam must be assigned to a room without exceed the room capacity.

$$\sum_{i=1}^{N} e_i.\beta_{if} \le L_f \quad \text{for all } f \in \tag{12}$$

Due to the complexity of the problem, constraints 6 and 10, could be relaxed if assigning an examination to multiple rooms is unavoidable (constraint 10) and it is not possible to assign the same room for students sitting consecutive exams in a day (constraint 6). Therefore, the exam has to be chunked, i.e. relaxing constraint 10.

As in standard benchmark dataset, wherever possible, examinations should be spread out over timeslots so that students have large gaps in between exams (soft constraint).

4. Objective Function

In order to adhere with practical issues, we introduce a new objective function (named as *Penalty Cost*) which is adapted from a proximity cost (proposed by Carter et al. [35] and Burke et al. [36, 37]), as follows:

$$\textbf{Minimise } F = \frac{\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} c_{ij}.Penalty\ (t_i, t_j)}{M} \tag{13}$$

where,

$$penalty\ (t_i, t_j) = \begin{cases} 2^{(5-\Delta t)(2-\Delta d)} & If\ |\Delta t| \le 5\ and\ |\Delta d| \le 2 \\ 0 & \text{otherwise;} \end{cases} \tag{14}$$

Equation 14 presents a weighted *penalty* value that reflect the cost of assigning exam E_i and E_j to timeslots. These being 0, 1, 2, 4, 8, 16, 64 and 256 where the cost is '0' if the gap of time slot for exam E_i and E_j is greater than 5 or the day gap is greater than 2. We only give a penalty up to a maximum of 5 timeslots in order to adhere with the well established proximity cost proposed by Carter et al. [35]. Whereas, we limit the penalty up to 2 days because 2 days gap between examinations gives ample free time for students (based on our pilot survey at UKM).

The new objective function (equations 13 and 14) aims to minimise the number of students having two exams in a row on the same day and tries to spread out exams over timeslots. Indeed, these formulations emphasize avoiding students having two consecutive exams on the same day instead of avoiding having two consecutive exams on different days, i.e. the penalty value for students having two consecutive exams on the same day (*penalty=256*) is higher than the penalty value for students having two consecutive exams on different days (*penalty=16*). This factor is not highlighted in the objective function proposed by Burke et al. [36, 37] and Carter et al. [35]. In fact, Carter et al. [35] totally ignores the day effect by assuming that the practical time gap between each consecutive timeslot is the same, each day has exam (during exam weeks), each day has the same number of time slots and the exams can be scheduled 24 hours a day without evening and weekend breaks. This can be observed based on their objective function and their standard benchmark datasets. The objective function introduced in Burke et al. [36] penalised consecutive exam on the same day. That is Burke et al. [36] only minimised the number of students having two consecutive exams on the same day without spreading exams over timeslots. Subsequently, Burke et al. [37] enhanced the objective function that was proposed in [36] by giving high penalty (3) for students having two consecutive exams on the same day and lower penalty (1) for two consecutive exams overnight. By adapting the three objective functions, we proposed a *Penalty Cost* that embeds the three features in one objective function.

5. Conclusions

In this paper, we have introduced a new real-world examination timetabling problem at the University Kebangsaan Malaysia with an objective to minimise student sitting consecutive exams on the same day by using a new proposed objective function, *Penalty Cost.* The *Penalty Cost* attempts to spread out exams over timeslots so that students have large gaps between exams and we emphasize on minimising consecutive exams on the same day. This work also enforces (hard constraint) no students sitting three consecutive exams on a day. This new objective function can also be applied to the standard benchmark examination datasets (Carter et al. [35]) by adding a new variable *day* for each corresponding timeslot. To adhere with the practical examination timetabling problem, we would also recommend adding weekend breaks and room capacity for each room into the standard benchmark examination datasets specification (Carter et al. [35]). The maximum seat capacity for each timeslot (on the standard benchmark examination datasets), have been applied by some researchers (see for example, Abdullah et al. [30] and Burke et al. [36]). Since the standard benchmark objective function for examination timetabling problems (proposed by Carter et al. [35]) was unable to cater for these new features of examination timetabling problem, we hope that future research in this area will consider our proposed objective function in evaluating the quality of generated timetables. The current standard objective function can still be applied for theoretical/preliminary work, but for solving the practical examination timetabling problems, our new objective function seem to be more appropriate.

Currently, we are designing and implementing a constructive heuristic which is adapted from a graph coloring heuristic to solve the UKM timetabling problem. Our future work will concentrate on scheduling examinations for a Law faculty which is slightly different with other faculties at the University Kebangsaan Malaysia. This faculty only has two slots per day (because the exam period is longer than other normal exams i.e. at 8:30am and 2:30pm) and exams have to be scheduled in a specific room only. We will also try to schedule invigilators and room assignments (which are done manually at this moment) and will incorporate it in this examination timetabling scheduler.

Acknowledgment

This work was supported by The University Kebangsaan Malaysia (UKM) with collaboration between Student Registry Department (user), Computer Center (solution provider) and ASAP research group, Nottingham (advisor). We wish to thank Nor Ashikin Baharom from Computer Center, UKM and Student Registry Department for fruitful discussion and contribution of UKM dataset.

References

[1] E.K. Burke, D.G. Elliman, P.H. Ford and R.F. Weare. (1996). Examination timetabling in British universities - A survey. The Practice and Theory of Automated Timetabling I: Selected Papers from 1st International Conference on the Practice and Theory of Automated Timetabling (PATAT I), Edinburgh, UK, Lecture Notes in Computer Science 1153, Springer-Verlag. (Editors: E.K. Burke and P. Ross), pp 76-92.

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

[3] S. Petrovic and E.K. Burke. (2004). University timetabling. Handbook of Scheduling: Algorithms, Models and Performance Analysis, Chapter 45, (Editor: J. Leung), CRC Press.

[4] E.K. Burke, J. Kingston and D. de Werra. (2004). Applications to timetabling. Handbook of Graph Theory, (Editors: J. Gross and J. Yellen), Chapman Hall/CRC Press, pp 445-474.

[5] N. Balakrishnan, A. Lucena and R.T. Wong. (1992). Scheduling examinations to reduce second order conflicts. Computers and Operations Research, 19, pp 353-361.

[6] G.M. White and P.W. Chan. (1979). Towards the construction of optimal examination schedules. INFOR, 17(3), pp 219-229.

[7] P. David. (1998). A constraint-based approach for examination timetabling using local repair technique. The Practice and Theory of Automated Timetabling II: Selected Papers from 2nd International Conference on the Practice and Theory of Automated Timetabling (PATAT II), Toronto, Canada, Lecture Notes in Computer Science 1408, Springer-Verlag. (Editors: E.K. Burke and M. Carter), pp 169-186.

[8] P. Boizumault, Y. Delon and L. Peridy. (1996). Constraint logic programming for examination timetabling. The Journal of Logic Programming, 26, pp 217-233.

[9] J. Thompson and K. Dowsland. (1998). A robust simulated annealing based examination timetabling system. Computers Operations Research, 25, pp 637-648.

[10] L. Di Gaspero and A. Schaerf. (2001). Tabu search techniques for examination timetabling. The Practice and Theory of Automated Timetabling III: Selected Papers from 3rd International Conference on the Practice and Theory of Automated Timetabling (PATAT III), Konstanz, Germany, Lecture Notes in Computer Science 2079, Springer-Verlag. (Editors: E.K. Burke and W. Erben), pp 104-117.

[11] G.M White and B.S Xie. (2001). Examination timetables and tabu search with longer-term memory. The Practice and Theory of Automated Timetabling III: Selected Papers from 3rd International Conference on the Practice and Theory of Automated Timetabling (PATAT III), Konstanz, Germany, Lecture Notes in Computer Science 2079, Springer-Verlag. (Editors: E.K. Burke and W. Erben), pp 85-103.

[12] L.T.G. Merlot, N. Boland, B.D. Hughes and P.J. Stuckey. (2003). A hybrid algorithm for the examination timetabling problem. The Practice and Theory of Automated Timetabling IV: Selected Papers from 4th International Conference on the Practice and Theory of Automated Timetabling (PATAT IV), Gent, Belgium, Lecture Notes in Computer Science 2740, Springer-Verlag. (Editors: E.K. Burke and P. De Causmaecker), pp 207-231.

[13] E.K. Burke and J.P Newall. (2003). Enhancing timetable solutions with local search methods. The Practice and Theory of Automated Timetabling IV: Selected Papers from 4th International Conference on the Practice and Theory of Automated Timetabling (PATAT IV), Gent, Belgium, Lecture Notes in Computer Science 2740, Springer-Verlag. (Editors: E.K. Burke and P. De Causmaecker), pp 195-206.

[14] S. Petrovic and Y. Bykov. (2003). A multi-objective optimisation technique for exam timetabling based on trajectories. The Practice and Theory of Automated Timetabling IV: Selected Papers from 4th International Conference on the Practice and Theory of Automated Timetabling (PATAT IV), Gent, Belgium, Lecture Notes in Computer Science 2740, Springer-Verlag. (Editors: E.K. Burke and P. De Causmaecker), pp 181-194.

[15] E.K. Burke, B. MacCarthy, S. Petrovic and R. Qu. (2001). Case-based reasoning in course timetabling: an attribute graph approach. The Proceeding of the 4th International Conference on Case-based Reasoning (ICCBR-2001), Springer Lecture Notes in Artificial Intelligence 2080, pp 90-104.

[16] E.K. Burke, S. Petrovic and R. Qu. (2006b). Case-based heuristic selection for timetabling problems. Journal of Scheduling, 9(2), pp 115-132.

[17] H. Asmuni, E.K. Burke and J.M. Garibaldi. (2004). A comparison of fuzzy and non-fuzzy ordering heuristics for examination timetabling. The Proceedings of the 5th International Conference on Recent Advances in Soft Computing 2004 (RASC2004), Nottingham, UK, pp 288-293.

[18] H. Asmuni, E.K. Burke, J.M. Garibaldi and B. McCollum. (2005a). Fuzzy multiple ordering criteria for examination timetabling. The Practice and Theory of Automated Timetabling V: Selected Papers from 5th International Conference on the Practice and Theory of Automated Timetabling (PATAT V), Pittsburg, USA, Lecture Notes in Computer Science 3616, Springer-Verlag, (Editors: E.K. Burke and M. Trick), pp 334-353.

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

[20] G. Kendall and N. Hussin. (2005a). An investigation of a tabu search based hyper-heuristic for examination timetabling. Multidisciplinary Scheduling: Theory and Applications, (Editors: G. Kendall, E.K. Burke, S. Petrovic and M. Gendreau), Springer, pp 309-328.

[21] E.K. Burke and J.P. Newall. (2004). Solving examination timetabling problems through adaptation of heuristic orderings. Annals of Operations Research, 129, pp 107-134.

[22] D.E. Joslin and D.P. Clements. (1999). Squeaky wheel optimisation. Journal of Artificial Intelligence Research, 10, pp 353-373.

[23] A. Schaerf. (1999a). A survey of automated timetabling. Artificial Intelligence Review, 13(2), pp 87-127.

[24] M.W. Carter and G. Laporte. (1996). Recent developments in practical examination timetabling. The Practice and Theory of Automated Timetabling I: Selected Papers from 1st International Conference on the Practice and Theory of Automated Timetabling (PATAT I), Edinburgh, UK, Lecture Notes in Computer Science 1153, Springer-Verlag. (Editors: E.K. Burke and P. Ross), pp 3-21.

[25] D. de Werra. (1985). An introduction to timetabling. European Journal of Operational Research, 19, pp 151-162.

[26] V.A. Bardadym. (1996). Computer-aided school and university timetabling: a new wave. The Practice and Theory of Automated Timetabling I: Selected Papers from 1st International Conference on the Practice and Theory of Automated Timetabling (PATAT I), Edinburgh, UK, Lecture Notes in Computer Science 1153, Springer-Verlag. (Editors: E.K. Burke and P. Ross), pp 22-45.

[27] E.K. Burke, K. Jackson, J.H. Kingston and R.F. Weare. (1997). Automated university timetabling: the state of the art. The Computer Journal, 40(9), pp 565-571.

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

[29] S. Abdullah, S. Ahmadi, E.K. Burke and M. Dror. (to appear 2006a). Investigating Ahuja-Orlin's large neighbourhood search approach for examination timetabling. Accepted for publication in OR Spectrum.

[30] S. Abdullah , S. Ahmadi, E.K. Burke, M. Dror and B. McCollum. (to appear 2006b). A tabu-based large neighbourhood search approach for the capacitated examination timetabling problem. Accepted for publication in the Journal of the Operational Research Society.

[31] M. Ayob, E.K. Burke and G. Kendall. (2006). An iterative re-start variable neighbourhood search for the examination timetabling problem. Extended Abstract. Proceedings of the 6th International Conference on the Practice and Theory of Automated Timetabling. 30th August-1st September 2006, Brno, The Czech Republic, pp 336-344.

[32] E.K. Burke and Y. Bykov. (2006). Solving exam timetabling problems with the flex-deluge algorithm. Extended Abstract. Proceedings of the 6th International

Conference on the Practice and Theory of Automated Timetabling. 30th August-1st September 2006, Brno, The Czech Republic, pp 370-372.

[33] E.K. Burke, B. McCollum, P. McMullan and R. Qu. (2006). Examination timetabling: A new formulation. Extended Abstract. Proceedings of the 6th International Conference on the Practice and Theory of Automated Timetabling. 30th August-1st September 2006, Brno, The Czech Republic, pp 373-375.

[34] H. Asmuni, E.K. Burke, M. Garibaldi and B. McCollum. (2006). A novel fuzzy approach to evaluate the quality of examination timetabling. Proceedings of the 6th International Conference on the Practice and Theory of Automated Timetabling. 30th August-1st September 2006, Brno, The Czech Republic, pp 373-375.

[35] M.W. Carter, G. Laporte and S.Y. Lee. (1996) Examination timetabling: algorithmic strategies and applications. Journal of the Operational Research Society Volume 47 Issue 3, 373-383.

[36] E.K. Burke, J.P. Newall and R.F. Weare (1996). A Memetic Algorithm for University Exam Timetabling. The Practice and Theory of Automated Timetabling I: Selected Papers from 1st International Conference on the Practice and Theory of Automated Timetabling (PATAT I), Edinburgh, UK, Lecture Notes in Computer Science 1153, Springer-Verlag. (Editors: E.K. Burke and P. Ross), pp 3-21.

[37] E.K. Burke and J.P. Newall. (1999). A multistage evolutionary algorithm for the timetable problem. IEEE Trans. Evolutionary Computation. 3, 63-74.

Appendix

The UKM06-1 dataset has four text files: UKM06-1.stu, UKM06-1.slt, UKM06-1.rom and UKM06-1.isl. UKM06-1.stu contains student enrollment data where each row has exam and each line represent one student, i.e. each line shows the exam enrolled by a student. UKM06-1.slt has time slot and day data where each line contains day and time slot ID, e.g. 2 4 shows day 2 and time slot 4. UKM06-1.rom has room data where each line represent one room with the first, second and third row are room name, room capacity and room group, respectively. Room group indicates that the room that has the same group ID can be logically merged (students overflow) if necessary. UKM06-1.isl shows the exams that cannot share room with any other exams. UKM06-1.lec has information about exam and lecturer (invigilator).