A Comparison of Fuzzy and Non-Fuzzy Ordering Heuristics for Examination Timetabling

Hishammuddin Asmuni, Edmund K. Burke and Jonathan M. Garibaldi
Automated Scheduling, Optimisation and Planning (ASAP) Research Group
School of Computer Science and Information Technology, University of Nottingham
Jubilee Campus, Wollaton Road, Nottingham, NG8 1BB, United Kingdom
e-mail: {hba, ekb, jmg}@cs.nott.ac.uk

Abstract: *The decision of 'appropriately' choosing the order in which examinations are scheduled when constructing an examination timetable can be seen as a process of ordering alternatives from a set in order to find a good solution. In this paper, a comparison is made between fuzzy ordering with single and linear combinations of ordering heuristics. It is found that fuzzy ordering leads to better solutions.*

Keywords*: examination timetabling, heuristic ordering, fuzzy reasoning, fuzzy ordering*

1 Introduction

Examination timetabling is of much interest and concern to academic institutions. The basic problem is to allocate a timeslot for all exams within a limited number of permitted timeslots in order to find a conflict free timetable. This assignment process is subject to 'hard' constraints which must be satisfied in order to get a feasible timetable, such as no student is required to sit two exams at the same time. In addition, it is also important to build a good quality examination timetable that considers not only the administration requirements, but also takes into account lecturers' and students' preferences. These requirements are generally considered as 'soft' constraints which are desirable (but not essential) to satisfy. As reported by Burke et al. [6], these requirements vary from one academic institution to another. As this task is time consuming and tedious to do manually, many attempts have been made over the last few decades to generate timetables automatically. With a large number of exams needing to be assigned to timeslots and a list of constraints needing to be satisfied, the search space for this problem is very large. While some work only focuses on finding good initial solutions by using 'constructive' algorithms, many others use an iterative search procedure to progressively improve the initial solutions. Various search techniques have been developed to find good solutions to exam timetabling problems. These include Local Search [8][14][20], Tabu Search [18][19], Simulated Annealing [24], Genetic Algorithms [7], Memetic Algorithm [10][11][12], and the Great Deluge algorithm [4]. The recent state of the art of exam timetabling is overviewed in a variety of papers (refer to [13][15][21][23]).

One of the earliest techniques implemented in finding good initial solutions is known as the sequential constructive algorithm. The main principle is that exams are ordered by certain heuristics before each exam is sequentially chosen to be assigned to a timeslot. This ordering represents how difficult it is to schedule the exams. The idea is that by assigning the most difficult exams first it is likely that we can avoid generating unfeasible solutions. Here, a feasible solution means that all exams are assigned to timeslots without violating any of the specified hard constraints. Many studies have been made into the best way to calculate the 'difficulty to schedule an exam'. Carter et al. [16] have shown that a single ordering heuristic can guide the search algorithm for a good solution compared to random selection. The work of Black [3] has examined the usefulness of incorporating constraint weights into measures used to generate both static and dynamic exams orderings for the same benchmark problems used by Carter. Burke and Newall [9] introduced an adaptive heuristic technique in which they start ordering by a certain heuristic and then alter that heuristic ordering to take into account the penalty that exams are imposing upon the timetable.

The aim of this paper is to investigate and analyze the potential of using fuzzy methodologies to perform simultaneous multiple ordering. A sequential constructive algorithm was implemented with a single ordering heuristic and multiple ordering, both by fuzzy reasoning and linear combination. The performance of various ordering heuristics was compared on a set of standard benchmark problems. Ordered Weighted Averages (OWA) by Yager [25] and fuzzy linear programming by Zimmermann [26] are closely related to this problem but any comparison between their methods and this work is out of the scope of this paper. The rest of the paper is organized as follows: Section 2 briefly describes the constructive algorithm and the fuzzy model used. Section 3 presents the experimental results and finally, Section 4 contain some concluding remarks.

2 Methods and Fuzzy Modeling

Graph colouring based ordering heuristics can be used to measure the difficulty to schedule an exam (see [5]). The following list describes the three ordering heuristics that were considered in this work.

(a) Largest Degree (LD) - number of other exams in conflict.
(b) Saturation Degree (SD) – number of clash free timeslots available.
(c) Largest Enrollment (LE) – number of students enrolled.

There are many potential ways in which exams could be ordered using various combinations of these three heuristics with the consequence that different solutions will be produced. Essentially, the exam ordering will have an impact on how the search algorithm will navigate through the search space.

2.1 The Sequential Construction Heuristic

The sequential construction heuristic proposed by Carter et al. [16], but with a modification in the backtracking process, was applied to construct a timetable once the exams had been ordered by the various ordering techniques. The algorithm used is detailed in Fig. 1.

```
Sort unscheduled exams using selected ordering heuristic;
k := number of exams to be inserted before recalculate the exams ordering
While there exist unscheduled exam
    For e := 1 to k
      Select an exam with the highest difficulty to be scheduled from unscheduled list;
      If found clash free timeslot
           Insert exam into the last timeslot with least penalty;
           Remove exam from unscheduled exam list;
      Else
           Find timeslots where exam can be inserted with minimum number of scheduled
           exams need to be removed from the timeslot;
           If found more than one slot with the same number of scheduled exams need to be
           removed
             Select a timeslot randomly from the candidate list of slots, ts;
           End if
           c :=  number of exam in timeslot ts that conflict with exam[u];
           For m := 1 to c
              Select exam[m];
              If found another timeslot with minimum cost to move exam[m]
                Move exam[m] to the timeslot;
              else
                Bump back exam[m] to unscheduled exam list;
              End if
           End for
           Insert exam to timeslot ts;
           Remove exam from unscheduled exam list;
      End If
    End For
   Sort unscheduled exams using selected ordering heuristic;
End while
```

Fig. 1. Pseudo code for sequential construction heuristic

Having calculated the exam weight that represents the difficulty of the exam to be scheduled, an exam is sequentially chosen from the descending order list and assigned to the timeslot that causes least penalty. In the case where no clash free timeslot is available, some of the already scheduled exams need to be reshuffled either by moving to different timeslots or by removing them from the timetable and rescheduling on the next iteration. The timeslot with the minimum number of exams that need to be 'bumped back' into the unscheduled exams list is chosen. If there is a tie, the algorithm will select the timeslot randomly from the candidate list. In order to reduce the computational time, the exam ordering is recalculated after every k exams are inserted (k was set to 5 in the experiments).

2.2 Cost Function

The widely used proximity cost function is implemented to measure the timetable quality. The maximum capacity for each timeslot is not taken into account. Only feasible timetables are accepted and the penalty function is utilized to try to spread out each student's schedule. If two exams scheduled for a particular student are t timeslots apart, the weight is set to $w_t = 2^{5-t}$ where $t \in \{1, 2, 3, 4, 5\}$. The weight is multiplied by the number of students that sit both of the scheduled exams. The average penalty per student is calculated by dividing the total penalty by total number of students. The following formulation was used (adopted from Burke et al. [4]):

$$\text{minimize} \quad \frac{\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} s_{ij} \, w_{(p_j - p_i)}}{T} \qquad \textbf{(1)}$$

where N is number of exams, s_{ij} is number of student enrolled both exam i and j, p_i is the timeslot where exam i is scheduled, p_j is the timeslot where exam j is scheduled, T is total number of students and subject to $1 \le p_j - p_i \le 5$.

2.3 Linear Multiple Ordering Heuristic

The order in which exams are scheduled is important in finding a good solution [3]. One way to simultaneously consider several ordering heuristics in measuring the exam difficulty weight is to multiply the value of the criteria for that exam with a weighting factor. When this method is used, the weighted function becomes, for example:

$$W(e_j) = w_d LD_j + w_e LE_j + w_s SD_j \qquad \textbf{(2)}$$

where $j = 1,2, \ldots N$; $w_d = w_e = w_s = \{0.0, 0.1, \ldots, 1.0\}$ if $N <= 400$; or $w_d = w_e = w_s = \{0.0, 0.25, 0.5, 0.75, 1.0\}$ if $N > 400$; and w_d, w_e, w_s are weighting factors for LD, LE and SD respectively.

2.4 Fuzzy Multiple Ordering Heuristic

In practice, the choice of 'appropriate' exam ordering always involves uncertainty. For example, it may be assumed that an exam is more difficult to schedule if it has a 'large' number of exams in conflict and a 'small' number of available slots. This is dealing with imprecise and vague information, where the exact values for 'large' and 'small' are not known with certainty. Hence, it would appear that this problem is one where fuzzy techniques may fit well.

A fuzzy expert system was designed in which two out of the three heuristics above or all three ordering heuristics are selected as input variables. Each of the input variables is assigned three linguistic terms; triangular shape fuzzy sets corresponding to meanings of *small*, *medium* and *high*. A restricted form of exhaustive search was implemented to find the most appropriate shape for the linguistic terms. The memberships function were tuned by altering the parameter *cp* which represents the right edge for the term *small*, the centre point for the linguistic label *medium* and the left edge for the term *high* as illustrated in Fig. 2. A search was then carried out to find the best set of *cp* parameters (there was one for each linguistic variable – i.e. a *cp* parameter for each of the input variables and the output variable). During the search for the 'optimal' fuzzy model, the centre point for any of the fuzzy variables might take a value between 0.0 and 1.0 (inclusive). Increments of value 0.1 were used for datasets that have 400 and fewer exams and increments of value 0.25 were used for datasets that have more than 400 exams.

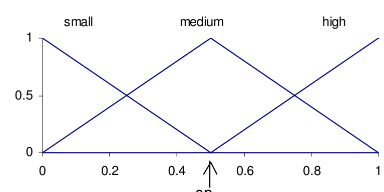

Fig. 2. Membership function for fuzzy variables

Different sets of fixed rulesets have been designed for different combinations of input variables. In the case where there are only two input variables involved, 9 fuzzy rules were implemented. Fig.3-5 show the rules used for input variables LD + LE, LE + SD and LD + SD respectively. Fig.6 illustrates the ruleset, consisting of 27 fuzzy rules, used for combining three ordering heuristics. Standard Mamdani style fuzzy inference was implemented with standard Zadeh (min-max) operators. Centroid defuzzification was utilized to obtain a single crisp (real) value for *exam_weight*.

		LE		
		S	**M**	**H**
LD	**S**	VS	S	M
	M	S	M	H
	H	M	H	VH

Fig. 3. Fuzzy ruleset for variables LD + LE

		SD		
		S	**M**	**H**
LE	**S**	M	S	VS
	M	H	M	S
	H	VH	H	M

Fig. 4. Fuzzy ruleset for variables LE + SD

		SD		
		S	**M**	**H**
LD	**S**	M	S	VS
	M	H	M	S
	H	VH	H	M

Fig. 5. Fuzzy ruleset for variables LD + SD

LE	**LD**								
	S			**M**			**H**		
	SD			**SD**			**SD**		
	S	**M**	**H**	**S**	**M**	**H**	**S**	**M**	**H**
S	S	VS	VS	S	S	VS	M	S	S
M	S	S	VS	H	M	M	H	M	M
H	H	S	S	H	M	M	VH	H	M

Fig. 6. Fuzzy ruleset for variables LD + SD + LE

3 Experimental Results

The system was implemented using java based object oriented programming, utilising the fuzzy inference engine developed by Sazonov et al. [22]. The experiments were run on a PC with a 1.8 GHz Pentium 4 and 256MB of RAM. Carter's publicly available exam timetabling datasets were used in the experiments [16]. As a control the single ordering heuristics were implemented by setting one of the weighting factors to the value 1.0 and the other two to 0.0 as per Eq. (2). Table 1 reproduces the problem characteristics and shows the results obtained for the single ordering heuristic.

Table 1. Characteristics of the problem and results for single ordering heuristic

No. of Student	No. of Exams	No. of Student	No. of Slots	Results for single ordering heuristic		
				LD	SD	LE
CAR-F-92	543	18419	32	4.85	6.45	4.76
CAR-S-91	682	16925	35	5.94	7.78	5.74
EAR-F-83	190	1125	24	41.21	48.40	43.94
HEC-S-92	81	2823	18	14.16	17.18	13.42
KFU-S-93	461	5349	20	18.37	21.38	16.30
LSE-F-91	381	2726	18	14.94	17.10	13.12
RYE-F-92	486	11483	23	13.33	15.32	11.97
STA-F-83	139	611	13	167.05	173.14	172.22
TRE-S-92	261	4360	23	10.40	11.68	9.46
UTA-S-92	622	21266	35	4.05	5.13	3.78
UTE-S-92	184	2750	10	33.67	34.77	28.89
YOR-F-83	181	941	21	45.30	52.23	42.60

Table 2 shows the results when implementing multiple ordering heuristics compared to Carter's sequential constructive algorithm. The last column in Table 2 shows the best results for the benchmark datasets compiled from [1][4][8][14][20]. It is obvious that fuzzy multiple ordering heuristic has outperformed the linear multiple ordering heuristic for all of the datasets. In comparison with the best result in single ordering heuristic, linear multiple ordering heuristic with the LD+SD combination produced worst results in 11 datasets; fuzzy multiple ordering heuristic with LD+SD combination produced the worst results in UTE-S-92 and YOR-F-83. When considering only two ordering heuristics, fuzzy multiple ordering heuristic with the SD+LE combination produced better results in 10 datasets. However, three multiple ordering heuristics outperformed the two multiple ordering heuristics in overall. Although our best results did not beat any of the best benchmark results, the fuzzy based ordering produced better results for CAR-F-92, CAR-S-91, STA-F-83, TRE-S-92 and YOR-F-83 than Carter's constructive approach. Due to space limitations, the fuzzy models that produced the best solutions for fuzzy based ordering shown in Table 2 are not presented here. Although tuning the membership functions shown in Fig. 2 required extra search effort, the results appear quite promising.

Table 2. Experimental results for the multiple ordering heuristic.

Dataset	LD + LE		SD + LE		LD + SD		LD + SD + LE		Our best results	Carter et al. [16]	Best results from [1][4] [8][14][20]
	Linear	Fuzzy	Linear	Fuzzy	Linear	Fuzzy	Linear	Fuzzy			
CAR-F-92	4.86	4.59	4.76	4.45	4.80	4.68	4.64	4.52	4.45	6.2	4.10
CAR-S-91	5.47	5.50	5.63	5.31	5.82	5.45	5.41	5.24	5.24	7.1	4.65
EAR-F-83	38.21	38.09	41.05	36.99	40.07	39.34	37.96	37.11	36.99	36.4	29.30
HEC-S-92	12.89	12.14	13.20	12.03	13.43	12.69	12.77	11.71	11.71	10.8	9.20
KFU-S-93	16.29	15.88	16.21	15.94	17.65	16.09	16.03	15.34	15.34	14	13.46
LSE-F-91	12.98	12.10	13.12	12.16	14.42	12.91	12.47	11.43	11.43	10.5	9.60
RYE-F-92	10.56	11.03	11.05	10.35	12.18	11.77	10.47	10.30	10.30	7.3	6.80
STA-F-83	167.05	159.82	167.61	159.29	167.05	165.31	167.05	159.15	159.15	161.5	150.28
TRE-S-92	9.26	8.99	9.26	8.90	9.99	9.26	9.21	8.64	8.64	9.6	8.13
UTA-S-92	3.68	3.72	3.75	3.55	4.00	3.73	3.57	3.55	3.55	3.5	3.20
UTE-S-92	28.68	28.65	28.66	27.91	32.94	29.69	28.42	27.64	27.64	25.8	24.21
YOR-F-83	42.43	41.21	42.22	40.71	43.81	43.00	41.97	40.68	40.68	41.7	36.11

4 Discussion and future research

In our previous work [2], it was demonstrated that multiple ordering heuristics, utilising fuzzy techniques to consider two ordering heuristics simultaneously, could outperform any single ordering heuristic in the benchmark datasets used. In this paper, these experiments have been extended by improving the construction algorithm and by utilising up to three ordering heuristics in the fuzzy expert system. For 10 out of the 12 datasets used, better results were obtained compared to two (fuzzy) ordering heuristics. This indicates that the selection of combinations of ordering heuristics is important in order to get good quality solutions. It is not the case, however, that three ordering heuristics always performed better than two ordering heuristics. For two datasets (CAR-F92 and EAR-F-83), two ordering heuristics (SD+LE) produced the best overall result. This is probably due to the fact that a fixed fuzzy rule set was implemented in each case – no tuning of fuzzy rules was implemented. If the rule set was tuned, then it should be possible to find a model based on three ordering heuristics to outperform that based on two (assuming that it is possible to search a reasonable proportion of the overall model search space). In addition, this study also confirms that, as might be expected, fuzzy reasoning does result in better solutions compared to linear combinations. Although fuzzy techniques required longer processing time, this is acceptable because once the best fuzzy model is known for the problem instances, the constructive algorithm can produce the solution in a reasonable time.

The main objective of this research was to investigate the effect of simultaneously considering multiple ordering heuristics when finding solutions for examination timetabling. The ordering represents the difficulty of the exam to be scheduled. How the exams are ordered and chosen sequentially will influence the behavior of the search algorithm in finding feasible solutions. Rather than employing the single ordering heuristic (which is usually used), this paper proposes a new approach to calculate exam difficulty by taking into account several ordering heuristics at the same time. Two approaches have been used to calculate the weight: linear combination and fuzzy reasoning.

As future work, the authors will be experimenting with constraint weight ordering heuristics as described in [3]. In the next stage, the authors aim to investigate iterative improvement utilising the Great Deluge Algorithm when it is started with the good initial solutions produced in this paper. Finally, the authors expect to explore the use of more sophisticated search algorithms in tuning the membership functions and the fuzzy rules.

Acknowledgements. This research work was supported by the Universiti Teknologi Malaysia (UTM) and the Ministry of Science, Technology and the Environment Malaysia (MOSTE).

5 References

[1] Abdullah, S, Ahmadi,,S., Burke, E. K. and Dror, M. Computer Science Technical Report *NOTTCS-TR-2004-8*, University of Nottingham. August 2004.

[2] Asmuni, H., Burke, E.K., Garibaldi, J. M. Fuzzy Multiple Ordering Criteria for Examination Timetabling. In: E. K. Burke & M. Trick (eds): *Proceedings of the 5th International Conference on Practice and Theory of Automated Timetabling (PATAT 2004)*, Pittsburgh, USA, August 2004, pages 51-65.

[3] Black, D.P. Search in Weighted Constraint Satisfaction Problems. *PhD Thesis*, University of Leeds, United Kingdom, 2003.

[4] Burke, E.K., Bykov, Y., Newall, J., Petrovic, S. A time-predefined local search approach to exam timetabling problems. *IIE Transactions on Operations Engineering* **36** (2004) 509-528.

[5] Burke, E.K., de Werra, D., Kingston, J.: Applications in timetabling. In: Yellen, J., Gross, J.L (Eds.): Handbook of Graph Theory. Chapman Hall, CRC Press. (2003) 445-474

[6] Burke, E.K., Elliman, D.G., Ford, P.H., Weare, R.F. Examination timetabling in British Universities – a survey. In: Burke, E, Ross, P. (Eds.): *Practice and Theory of Automated Timetabling I (PATAT 1995, Edinburgh, Aug/Sept, selected papers). Lecture Notes in Computer Science*, Vol. 1153. Springer-Verlag, Berlin Heidelberg New York (1996) 76-90.

[7] Burke, E.K., Elliman, D.G. and Weare, R.F. A hybrid genetic algorithm for highly constrained timetabling problems. *Proceedings of the 6th International Conference on Genetic Algorithms* (ICGA'95, Pittsburgh, USA, 15th-19th July 1995). (1995) 605-610, Morgan Kaufmann, San Francisco, CA, USA.

[8] Burke, E.K., Newall, J.P. Enhancing Timetable Solutions with Local Search Methods. In: Burke, E, Causmaecker, P.D. (Eds.): Practice and Theory of Automated Timetabling IV (PATAT 2002, Gent Belgium, August, selected papers). Lecture Notes in Comp. Science, Vol. 2740. Springer-Verlag, Berlin Heidelberg New York (2003) 195-206.

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

[10] Burke, E. K. and Newall, J.P. A Multi-Stage Evolutionary Algorithm for the Timetable Problem, the *IEEE Transactions on Evolutionary Computation*, **3.1**, (1999) 63-74.

[11] Burke, E. K., Newall, J.P. and Weare, R.F. A Memetic Algorithm for University Exam Timetabling, *The Practice and Theory of Automated Timetabling (eds EK Burke and P Ross), Lecture Notes in Computer Science* Vol. 1153, Springer (1996) 241-250.

[12] Burke, E. K., Newall, J.P. and Weare, R.F. Initialisation Strategies and Diversity in Evolutionary Timetabling, *Evolutionary Computation Journal* (special issue on Scheduling), **6** (1998) 81-103.

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

[14] Caramia, M., Dell'Olmo, P., Italiano, G.F. New algorithms for examination timetabling. In: Naher, S., Wagner, D. (Eds.): *Algorithm Engineering 4th Int. Workshop, Proc. WAE 2000 (Saarbrucken, Germany, September) Lecture Notes in Computer Science,* Vol. 1982. Springer-Verlag, Berlin Heidelberg New York (2001) 230-241.

[15] Carter, M.W., Laporte, G. Recent developments in practical examination timetabling. In: Burke, E., Ross, P. (Eds.): Practice and Theory of Automated Timetabling I (PATAT 1995, Edinburgh, Aug/Sept, selected papers). Lecture Notes in Computer Science, Vol. 1153. Springer-Verlag, Berlin Heidelberg New York (1996) 3-21.

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

[17] Cote, P., Wong, T., and Sabourin, R. Application of a Hybrid Multi-Objective Evolutionary Algorithm to the Uncapacitated Exam Proximity Problem. In: E.K. Burke & M. Trick (eds): *Proceedings of the 5th International Conference on Practice and Theory of Automated Timetabling (PATAT 2004),* Pittsburgh, USA, August 2004, pages 151-167.

[18] Di Gaspero, L., Schaerf, A. Tabu search techniques for examination timetabling. In: Burke, E., Erben, W. (Eds.): Practice and Theory of Automated Timetabling III (PATAT 2000, Konstanz Germany, August, selected papers). Lecture Notes in Computer Science, Vol. 2079. Springer-Verlag, Berlin Heidelberg New York (2001) 104-117.

[19] Kendall, G and Hussin, N. M. Tabu Search Hyper-Heuristic Approach to the Examination Timetabling Problem at University Technology MARA. In: E. K. Burke & M. Trick (eds): *Proceedings of the 5th International Conference on Practice and Theory of Automated Timetabling (PATAT 2004)*, Pittsburgh, USA, August 2004, pages 199-217.

[20] Merlot, L.T.G., Boland, N., Hughes, B.D., Stuckey, P.J. A hybrid algorithm for examination timetabling problem. In: Burke, E, Causmaecker, P.D. (Eds.): *Practice and Theory of Automated Timetabling IV (PATAT 2002, Gent Belgium, August, selected papers). Lecture Notes in Computer Science,* Vol. 2740. Springer-Verlag, Berlin Heidelberg New York (2003) 207-231.

[21] Petrovic, S., Burke, E.K. University Timetabling, Ch. 45 in the *Handbook of Scheduling: Algorithms, Models, and Performance Analysis* (ed. J. Leung), Chapman and Hall/CRC Press, (2004).

[22] Sazonov, E. S., Klinkhachorn, P., Gangarao, H.V.S., Halabe, U.B. Fuzzy logic expert System for automated damage detection from changes in strain energy mode shapes. *Nondestructive Testing and Evaluation*. **18**(1) (2002) 1-20.

[23] Schaerf, A. A survey of automated timetabling. Artificial *Intelligent Review*. **13** (1999) 87-127

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

[25] Yager, Y.Y. On ordered weighted averaging aggregation operators in multi-criteria decision making, *IEEE Trans. Systems Man Cybernet*. 18 (1988) 183-190.

[26] Zimmermann, H.J. Fuzzy programming and linear programming with several objective functions. *Fuzzy Sets and Systems.* 1(1) (1978) 45-55.