ANALYSIS OF BACKTRACKING IN UNIVERSITY EXAMINATION SCHEDULING

Siti Khatijah Nor Abdul Rahim[1,2]
[1]School of Computer Science
University of Nottingham
Jalan Broga, 43500, Semenyih
Selangor, Malaysia
[2]Universiti Teknologi MARA
32610, Bandar Baru Seri Iskandar
Perak, Malaysia

Andrzej Bargiela[3,4]
[3]School of Computer Science
University of Nottingham
Jubilee Campus, Wollaton Road,
Nottingham, NG8 1BB UK,
[4]Institute of Informatics,
Cracow Technical University
Poland

Rong Qu[3]
[3]School of Computer Science
University of Nottingham
Jubilee Campus, Wollaton Road,
Nottingham, NG8 1BB UK

KEYWORDS
Examination Scheduling, Backtracking, Fewer Slots.

ABSTRACT

Simulation modelling of the initial assignments of exams to time-slots provides an alternative approach to the establishment of a set of feasible solutions that are subsequently optimized. In this research, we analyze two backtracking strategies for reassigning exams after the initial allocation of exams to time-slots. We propose two approaches for backtracking, BT1 and BT2. The study indicates that backtracking is an effective approach for improving the quality of the examination schedule where BT2 has outperformed BT1 in a number of cases.

INTRODUCTION

Examination timetabling or scheduling is a process of creating feasible examination schedules with the objective to satisfy all hard constraints and some other soft constraints. There are many approaches proposed in the literature to solve this timetabling problem.

In many examination timetabling problems, satisfying the requirement about the specific number of timeslots in the given examination session is a hard constraint and is frequently quite a challenging task. The methods that generate examinations schedules using an arbitrary number of time-slots are much easier to design and implement but the solutions are clearly unacceptable as the final schedules.

FRAMEWORK OF THE PROPOSED APPROACH

In our previous work (Rahim et. al, 2012), (Rahim et. al, 2013) we have proposed a method to solve the timetabling problems which consists of 1) pre-processing, 2) a two-stage scheduling and 3) timetable optimization.

During scheduling process, the order of processing of exams may sometimes lead to non-optimal assigning of exams to slots which could create an infeasible schedule (i.e.: does not satisfy the minimum number requirement of slots). This situation calls for a reassigning of exams from the initial slot allocation to other slots in order to ensure the number of slots is reduced to the required number and the schedule becomes feasible. Logically, this kind of reassignment will need to relook or backtrack the initial allocation or assignment process, and therefore we will call this a *backtracking* process. In the backtracking process, some assignments already made will be undone in order to schedule these exams in other time-slots. As a result, this simulates a generation of as set of feasible schedules that will be used in the optimization process later.

The objectives of the *backtracking* might include 1) to reduce the number of slots in order to satisfy the minimum number requirement of slots in a given problem; 2) to prepare the non-optimal schedule for further optimization. In this paper, our objective is the latter. Another objective would be to get the lowest number of slots in order to minimize the duration of the examination session.

This is in anticipation that by reducing the number of slots at the early stage, one can minimize the cost of timetables at the later stage during the optimization process. The initial schedule with a few slots (i.e.: less than required number of slots), can always be modified into one with the required slots. We hypothesize that this could provide a useful buffering space during the optimization involving permutations of exams slots.

Proceedings 27th European Conference on Modelling and Simulation ©ECMS Webjørn Rekdalsbakken, Robin T. Bye, Houxiang Zhang (Editors)
ISBN: 978-0-9564944-6-7 / ISBN: 978-0-9564944-7-4 (CD)

Consequently, this has a potential for improving the quality of the schedules (Rahim et. al, 2009), (Rahim et. al, 2012).

It is important to highlight here that the cost of the schedule will be evaluated by the objective function proposed by (Carter et. al, 1996) as follows:

$$\frac{1}{T}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} s_{ij} w_{|pj - pi|} \qquad (1)$$

where N is the number of exams, S_{ij} is the number of students enrolled in both exam i and j, p_j is the time slot where exam j is scheduled, p_i is the time slot where exam i is scheduled, W is the cost imposed on the timetable for students sitting two exams $| p_j - p_i |$ slots apart, where W_1 =16, W_2=8, W_3=4, W_4=2 and W_5=1 and T is the total number of students. According to this cost function, a student taking two exams that are $| p_j - p_i |$ slots apart, where $| p_j - p_i | =\{1, 2, 3, 4, 5 \}$, leads to a cost of 16, 8, 4, 2, and 1, respectively. The objective of this cost function is to minimize the sum of costs per student. The lower the cost obtained, the better is the quality of the schedule, since the gap between two consecutive exams allows students to have additional revision time.

Figure 1 illustrates the backtracking phase in our general framework (Rahim et. al, 2012), (Rahim et. al, 2013) proposed in solving the examination timetabling problem. Note that the backtracking process is proposed to be done right after the initial assignment of exams to slots where both are parts of the scheduling process. The scheduling process is prior to the pre-processing stage and later an optimization stage will follow.

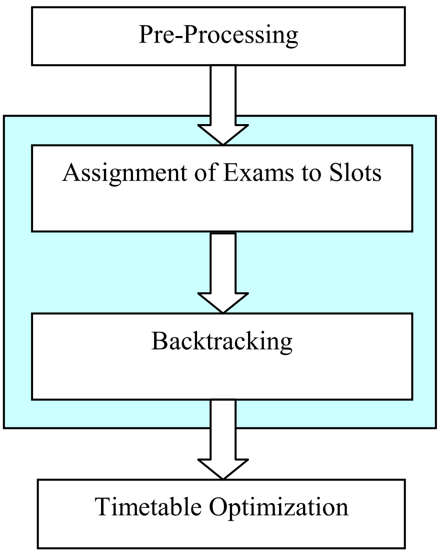

Figure 1: Scheduling Framework With Backtracking

In this study, we use two types of backtracking, as explained in detail in the next section. The pre-processing stage uses the information abstraction ideas as described in detail in (Bargiela et. al. 2002, 2008) and (Pedrycz et. al, 2000).

BACKTRACKING

First Method: Backtracking 1 (BT1)

In the first backtracking method, called here Backtracking 1 (BT1), we attempt to eliminate the last utilized time-slot.

We have implemented the backtracking process by (Carter et. al, 1996) but with some modifications. In contrast to (Carter et. al, 1996) where backtracking was performed during the initial placement of exams, in our approach, the placement of exams to their allocated slots has already been completed therefore we are attempting to convert the infeasible schedule into a feasible one.

After allocations of exams to slots were done, we identified all the exams in the last slot and we assigned them to a waiting list of unscheduled exams. Then, for each exam in this list, we initialized the selection criteria which is known as *Bp* (according to Carter et. al, 1996) for all periods equals to zero (*Bp=0*). Next for each of the exam in the list we proceed by finding the numbers of exams clashing with it in each of the available periods. *Bp* for each period is the number of exams clashing with the exam currently being evaluated in the waiting list. Please note that the exams clashing with the exam in the list are the exams that will be bumped to the waiting list, and thus will be assumed as unscheduled exams. (Note: we process the exams in the list on a First In First Out basis).

By contrast to (Carter et. al, 1996), we have assigned *Bp* equals to *number of exams + 1* (*Bp= nex + 1*) if the exam in the list has bumped any clashing exams encountered in the period we are dealing with. We also assign *Bp = nex+1* for a period, if the exam in the list originated from this period. This is another modification done to Carter's method to avoid cyclic. We continue finding the *Bp* for all periods for each exam in the waiting list.

The purpose of finding the Bp's for all the periods is to determine which period to choose to assign the exams in the waiting list. Bp's that we obtained for all periods can range from the value of *0* to *nex + 1*. So, the best *Bp* would be *0* and the worse *Bp* would be *nex + 1*. This means that, the exam in the waiting list will be assigned to the period with the minimum value of *Bp*.

In the period selection stage, there is always a possibility of having the same Bp's values. If there are a few periods having *Bp = 0*, then our method will choose the first period with *Bp=0* encountered, or in other words, the first available period with no exams clashing with the exam in the waiting list. In cases, where *Bp* range from the value 1 to nex (*Bp=1* to *Bp=nex*), and there exist multiple periods with the same Bp's, then our method will do a selection based on weighing given to the periods.

The weighing given was based on the total number of students having conflicts in both exams in the periods and the exam in the waiting list. The period with the maximum value of the weighing will be selected, thus the clashing exams in the period with the exam in the waiting list will be bumped to the waiting list. The weighing given is mainly for the purpose of breaking the ties of the same Bp's.

Once the period or the location to assign the exam in the waiting list is determined, the transfer stage follows. Transfer stage is the process of transferring the current exam in the waiting list to the new period selected.

The above process then repeats for other exams in the waiting list. If at the end of the process, some exams fail to be assigned to any periods, then we assume the backtracking process fails, thus the above process will be undone and the previous configurations of allocation of exams to periods will be used.

Second Method: Backtracking 2 (BT2)

In the second backtracking approach (BT2), the objective is to eliminate the slot containing the fewest number of exams after the allocation method. The number of slot that will be eliminated is also 1 (same as BT1).

It is interesting to note here that, in BT2, the slot that will be eliminated could be any slot in the schedule (in BT1 it is always the last slot), therefore it could be the first, in the middle or the last one.

Once the slot with the fewest exams has been determined, all the exams will be put in a waiting list. Each exam in the list will be evaluated for reallocation as per our first approach (BT1).

RESULTS AND DISCUSSIONS

We have evaluated our both approaches of backtracking on the randomly generated benchmark dataset used for the evaluation of examination timetabling algorithms. The dataset can be downloaded from (http://www.cs.nott.ac.uk/~rxq/data.htm).

The benchmark exam timetabling problem dataset consists of 18 different problem instances; 9 small (< 100 exams) and 9 large problems (>= 500 exams). The problems generated have conflict density values from 6% to 47% using 5% intervals. The number of students and their enrolments are variable according to the problem size and conflict density. The problems use the same objective function as in (1).

In the interest of clarity of presentation we used in this research, the 9 small problems. The characteristics of the problems are given in Table 1.

Table 1: Characteristics of the Randomly Generated Problems (Small Problems)

(a) Name of Dataset; (b) No of Exams; (c) No of Students; (d) No of Enrollments; (e) Conflict Density (f) Required No of Slots;

(a)	(b)	(c)	(d)	(e)	(f)
SP5	80	66	194	7%	15
SP10	100	100	359	11%	15
SP15	80	81	314	17%	15
SP20	80	83	344	19%	15
SP25	80	119	503	26%	15
SP30	80	126	577	32%	15
SP35	100	145	811	36%	19
SP40	81	168	798	42%	19
SP45	80	180	901	47%	19

Table 2: Results Obtained by BT1

(a)Name of Dataset; (b) No of Slots (original allocation); (c) No of Slots (after backtracking); (d) Cost (after backtracking); (e) No of Slots (Permutation 1); (f) Cost (after Permutation); (g) No of Slots (after added slots); (h) Cost (after Permutation - after added slots)

(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)
SP5	7	7	31.85	7	19.59	15	4.12
SP10	10	10	45.06	10	25.19	15	12.18
SP15	9	9	53.07	9	39.78	15	16.68
SP20	10	10	53.75	10	39.96	15	20.3
SP25	13	12	46.39	12	34.87	15	25.48
SP30	13	13	51.03	13	41.83	15	33.62
SP35	19	18	58.63	18	50.94	19	47.5
SP40	17	17	44.59	17	34.18	19	29.01
SP45	18	17	48.78	17	36.82	19	31.76

Table 3: Results Obtained by BT2

(a)Name of Dataset; (b) No of Slots (original allocation); (c) No of Slots (after backtracking); (d) Cost (after backtracking); (e) No of Slots (Permutation 1); (f) Cost (after Permutation); (g) No of Slots (after added slots); (h) Cost (after Permutation - after added slots)

(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)
SP5	7	7	31.85	7	19.59	15	4.12
SP10	10	10	45.06	10	25.19	15	12.18
SP15	9	9	53.07	9	39.78	15	16.68
SP20	10	10	53.75	10	39.96	15	20.3
SP25	13	12	44.1	12	31.34	15	25.17
SP30	13	13	51.03	13	41.83	15	33.62
SP35	19	18	58.63	18	50.94	19	47.5
SP40	17	17	44.59	17	34.18	19	29.01
SP45	18	17	47.04	17	34.25	19	31.12

Table 4: Comparison of Results by BT1, BT2 and Without Backtracking (W/O) on Dataset SP25, SP35 and SP45.

(a)Type of Experiment; (b) No of Slots (original allocation); (c) No of Slots (after backtracking); (d) Cost (after backtracking); (e) No of Slots (Permutation); (f) Cost (after Permutation); (g) No of Slots (after added slots); (h) Cost (after Permutation - after added slots)

(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)
SP25							
BT1	13	12	46.39	12	34.87	15	25.48
BT2	13	12	44.1	12	31.34	15	**25.17**
W/O	13	NA	NA	NA	NA	15	25.33
SP35							
BT1	19	18	58.63	18	50.94	19	47.5
BT2	19	18	58.63	18	50.94	19	47.5
W/O	19	NA	NA	NA	NA	19	**46.88**
SP45							
BT1	18	17	48.78	17	36.82	19	31.76
BT2	18	17	47.04	17	34.25	19	**31.12**
W/O	18	NA	NA	NA	NA	18	31.31

BT1 – Backtracking 1
BT2 – Backtracking 2
W/O – Without Backtracking
NA – Not Applicable

For each backtracking approach, we have recorded the number of slots and cost obtained after performing the backtracking (with the reduced number of slots), and later we recorded the results after doing permutations of exams slots (a type of optimization – (Rahim et. al, 2012)). The schedule for each dataset then has been added a number of slots to satisfy the requirement given in the problem (15 slots for the first 6 problems, and 19 slots for the balance). Later, the permutations of exams slots on these schedules (satisfying the requirement number of slots) were repeated to obtain the new cost. All the results by performing the mentioned steps here can be seen in Table 2 and Table 3. Based on these results, it can be seen that 3 datasets SP25, SP35 and SP45 (coincidently the same datasets) managed to reduce the slot after BT1 and BT2. The original number of slots before both backtracking for SP25 is 13 and after is 12, SP35 is 19 and 18; and SP45 is 18 and 17.

In terms of the cost obtained in Table 4, it can be seen that BT2 has outperformed BT1 in 2 datasets SP25 and SP45 (out of the 3 datasets) where lower costs were obtained after the optimization. To evaluate the advantage of the backtracking, we also have presented the results for these 3 datasets if backtracking is not performed. Again, for SP25 and SP45, BT2 has outperformed the ones without backtracking in terms of the costs obtained.

One possible reason why BT2 outperformed BT1 in the cases discussed above, is maybe because BT2 selected the slot with the fewest exams to be eliminated, and therefore only a few exams need to be assigned to other slots which indirectly means that only a few exams will need to be bumped out for further processing. By contrast to this, BT1 always selected the last slot, which does not guarantee that it is the slot containing the fewest exams. If this last slot contains many exams, therefore we can predict that it might involve more exams to be bumped out for further processing (as opposed to BT2).

For SP35, the results for both BT1 and BT2 are the same. This is because, the slot being selected for elimination by both BT1 and BT2 is coincidently the same slot which is the last slot. This is due to the fact that the last slot (selected by BT1) happens to be the same slot selected by BT2 where it has the fewest number of exams.

Figure 2 and Figure 3 (on page 5) illustrate the data structure of slots containing exams before and after backtracking for SP35 respectively. The first column in both Figure 2 and Figure 3 indicates the number of exams in the existing slot (each row). The number(s) in each row (starting from column 2) is the list of the exams allocated to the given slot.

As can be seen, before backtracking, the last slot has been assigned the fewest number of exams (Exam 30 and Exam 49). If we observe carefully, after backtracking, the number of slots has reduced by 1 (only 18 rows exist which represents 18 slots). Another important point to note is that, the ordering of some exams in the slots after backtracking has changed. This is due to the fact that assignment of exams from the waiting list to some slots will bumped out other exams to the waiting list and therefore these affected exams will sometimes be assigned to different slots (other than the initial original slots).

11	19	22	27	47	61	62	76	77	83	98	100
7	4	23	38	45	59	87	91	0	0	0	0
11	9	40	55	56	58	65	68	79	84	89	99
9	18	26	51	52	60	75	80	90	92	0	0
4	16	29	36	67	0	0	0	0	0	0	0
7	37	42	44	73	78	86	95	0	0	0	0
4	1	20	48	50	0	0	0	0	0	0	0
4	2	6	8	72	0	0	0	0	0	0	0
5	11	14	41	66	82	0	0	0	0	0	0
5	13	34	57	74	93	0	0	0	0	0	0
4	7	21	70	85	0	0	0	0	0	0	0
4	35	53	88	96	0	0	0	0	0	0	0
4	3	12	25	32	0	0	0	0	0	0	0
5	5	33	43	46	71	0	0	0	0	0	0
3	10	64	94	0	0	0	0	0	0	0	0
5	15	28	54	81	97	0	0	0	0	0	0
3	17	31	69	0	0	0	0	0	0	0	0
3	24	39	63	0	0	0	0	0	0	0	0
2	30	49	0	0	0	0	0	0	0	0	0

Figure 2: The Data Structure Illustrating the Exams Assignment to Slots Before Backtracking

12	17	24	27	61	62	76	77	79	83	94	98	100
7	20	23	38	45	53	56	87	0	0	0	0	0
9	5	39	48	55	68	84	89	90	99	0	0	0
9	29	33	47	52	60	74	80	91	92	0	0	0
5	18	26	51	75	85	0	0	0	0	0	0	0
8	10	30	40	42	44	65	73	95	0	0	0	0
4	15	43	46	97	0	0	0	0	0	0	0	0
6	1	16	28	49	64	67	0	0	0	0	0	0
6	11	14	41	59	66	82	0	0	0	0	0	0
2	9	37	0	0	0	0	0	0	0	0	0	0
5	13	34	57	72	93	0	0	0	0	0	0	0
6	3	32	35	36	50	96	0	0	0	0	0	0
3	2	8	70	0	0	0	0	0	0	0	0	0
4	31	54	78	86	0	0	0	0	0	0	0	0
3	4	12	88	0	0	0	0	0	0	0	0	0
3	7	71	81	0	0	0	0	0	0	0	0	0
4	19	22	58	69	0	0	0	0	0	0	0	0
4	6	21	25	63	0	0	0	0	0	0	0	0

Figure 3: The Data Structure Illustrating the Exams Assignment to Slots After Backtracking

An interesting point to note based on the results is that for SP35, the scheduling without doing backtracking actually produced better result which is 46.88 compared to BT1/BT2 which is 47.5. This might be due to the elimination of the slots via backtracking has resulted in changing of the initial assignment of exams to slots (through the allocation method) which disturbed the good ordering of exams generated earlier (i.e.. exams spaced out equally).

CONCLUSION

We conclude that the combined scheduling (that does not pay regard to the required number of time-slots) and backtracking (aimed at achieving the required number of time-slots) is an effective approach to examinations timetabling. However, in certain cases, the time-slot-aware scheduling without backtracking could give lower cost schedules because the backtracking can sometimes disturb the original ordering of exams to slots which might already be allocated to slots in an optimal way.

Comparing the two backtracking methods proposed here, we conclude that the second approach (BT2) has outperformed the first approach (BT1) which selected the slot with fewest exams for elimination.

REFERENCES

Abdul-Rahman, S., Burke E.K., Bargiela A., McCollum B., Ozcan E. "A Constructive approach to examination timetabling based on adaptive decomposition and ordering". Annals of Operation Research, 1-19, 2011, doi:10.1007/s10479-011-0999-8.

Abdul-Rahman, S., Bargiela A., Burke E.K., Ozcan E., McCollum B. "Linear combination of heuristic orderings in construction of examination timetable". European Journal of Operational Research , 2011, (published online, Oct. 2011).

Bargiela, A., and Pedrycz, W. 2002. "Granular Computing – An Introduction". Kluwer Academic Publishers. 2002. (doi: 10.1007/978-1-4615-1033-8).

Bargiela, A. and Pedrycz, W. 2008. "Toward a theory of Granular Computing for human-centred information processing". IEEE Trans. On Fuzzy Systems. 16(2): 320-330. (doi:10.1109/TFUZZ.2007.905 912).

Carter M., Laporte G. and Lee S. 1996. "Examination Timetabling: Algorithmic Strategies and Applications". Journal of Operations Research Society, 47 373-383.

Pedrycz W, Smith M.H., Bargiela A. 2000. "Granular signature of data". Proc. 19th Int. (IEEE) Conf.NAFIPS'2000, Atlanta. July 2000; 69-73. (doi: 10.1109/NAFIPS.2000.877387).

Rahim, S. K. N. A., Bargiela, A., & Qu, R. 2009. "Granular Modelling Of Exam To Slot Allocation". ECMS 2009 Proceedings edited by J. Otamendi, A. Bargiela, J. L. Montes, L. M. Doncel Pedrera (pp. 861-866). European Council for Modeling and Simulation. doi:10.7148/2009-0861-0866.

Rahim, SKNA., Bargiela, A., Qu, R. "Domain transformation approach to deterministic optimization of examination timetables". Artificial Intelligence Research, 2(1), 2013, (doi:10.5430/air.v2n1p122).

AUTHOR BIOGRAPHIES

SITI KHATIJAH NOR ABDUL RAHIM received her Bachelor's Degree in Computer Science, majoring in Software Engineering, from Universiti Sains Malaysia (USM), Penang, Malaysia in 2001. She later pursued her Master's Degree at the same university in Computer Science majoring in Computational Intelligence. She started her career as a full time lecturer with Universiti Teknologi Mara, Perak, Malaysia after receiving her Master's Degree in 2003. Currently she is in the final stage of her PhD study at the University of Nottingham Malaysia Campus. Her research interests lie in the area of Computational Intelligence, Granular Computing and Scheduling. Her email address is khyx8skn@nottingham.edu.my or sitik781@perak.uitm.edu.my.

ANDRZEJ BARGIELA is Professor in the School of Computer Science at the University of Nottingham. He served as President of the European Council for Modelling and Simulation (ECMS) during 2002-2006 and 2010-2012. He is Associate Editor of the IEEE Transactions on Systems Man and Cybernetics and Associate Editor of the Information Sciences. His research involves investigation into Granular Computing, human-centred information processing as a methodological approach to solving large-scale data mining. His email address is Andrzej.Bargiela@nottingham.ac.uk.

DR RONG QU is a Lecturer in the School of Computer Science at the University of Nottingham. She gained her PhD in Computer Science from the University of Nottingham in 2002. Her main research areas include meta-heuristics, constraint programming, IP/ILP, case based reasoning methodologies and knowledge discovery techniques on scheduling, especially educational timetabling, healthcare personnel scheduling, network routing problems and graph colouring. In total she has more than 30 papers published or to appear at international journals and peer-reviewed international conferences. Dr Qu is also a guest editor for special issues at the journal of Memetic Computing and the Journal of Scheduling, and the program chair of several workshops and an IEEE symposium. Her email address is rxq@cs.nott.ac.uk.