Roulette wheel Graph Colouring for Solving Examination Timetabling Problems

Nasser R. Sabar[1], Masri Ayob[2], Graham Kendall[3] and Rong Qu[4]

[1, 2] Jabatan Sains Komputer, Fakulti Teknologi dan Sains Maklumat,
Universiti Kebangsaan Malaysia, 43600 UKM, Bangi Selangor.
naserdolayme@yahoo.com , masri@ftsm.ukm.my
[3,4]ASAP Research Group, School of Computer Science, The University of Nottingham,
Nottingham NG8 1BB, UK.
gxk@cs.nott.ac.uk , rxq@cs.nott.ac.uk

Abstract. This work presents a simple graph based heuristic that employs a roulette wheel selection mechanism for solving examination timetabling problems. We arrange exams in non-increasing order of the number of conflicts (degree) that they have with other exams. The difficulty of each exam to be scheduled is estimated based on the degree of exams in conflict. The degree determines the size of a segment in a roulette wheel, with a larger degree giving a larger segment. The roulette wheel selection mechanism selects an exam if the generated random number falls within the exam's segment. This overcomes the problem of repeatedly choosing and scheduling the same sequence of exams. We utilise the proposed Roulette Wheel Graph Colouring heuristic on the un-capacitated Carter's benchmark datasets. Results showed that this simple heuristic is capable of producing feasible solutions for all 13 instances.

Keywords: scheduling**,** examination timetabling, graph colouring heuristics, roulette wheel selection.

1 Introduction

Examination timetabling problems deal with assigning a given set of exams into a given set of timeslots, subject to sets of hard and soft constraints [1]. A timetable is feasible if all exams have been assigned to timeslots and all required hard constraints are satisfied. Soft constraints are represent features that we wish to avoid but we can violate them if necessary. However, violations of soft constraints should be minimized as much as possible, and it is the minimization of the soft constraints that indicates the quality of the generated timetable.
Various approaches have been used to construct examination timetables such as graph colouring [1,2], fuzzy logic [3], ant algorithms [4] and neural networks [5]. Graph colouring heuristics represent exams as vertices and conflicts between exams as edges. The aim is to colour adjacent vertices with a different colour. Every colour represents a time period in the timetable. In the literature, various heuristics have been used to construct a conflict free timetable by using graph colouring models

[1,7,22,23,26]. These heuristics prioritize exams according to the level of scheduling difficulty. The rationale behind this is to make sure the most difficult exams are scheduled first. Examples of the most common graph colouring/timetabling heuristics are:

- Largest Degree First (LD): The exams are ordered according to the number of conflicts, in decreasing order.
- Least Saturation Degree First (LS): Exams are dynamically ordered by the number of available timeslots, in ascending order.
- Largest Enrolment First (LE): The exams are ordered according to the number of students enrolled, in decreasing order.

Asmuni et al. [3] used fuzzy logic to order exams based on graph colouring heuristics. When ordering the exams, by their degree of difficulty, fuzzy functions were introduced to evaluate the degree of difficulty. Results show that the fuzzy approach is capable of producing good quality solutions (when tested on Carter's benchmark datasets). However, they noted that different fuzzy functions need to be used on different problems to obtain the best results.

Corr et al. [5] utilized a neural network to determine the level of difficulty of assigning exams, with the most difficult exam being scheduled first. The neural network was built by storing feature vectors using three graph heuristics. The research demonstrated the feasibility of employing neural network based methods as an adaptive and generally applicable technique to timetabling problems.

More recently, Eley [4] applied an ant algorithm to simultaneously construct and improve the timetables. He used two ant colony approaches; MMAS-ET that is based on Max-Min Ant System (MMAS) (used by Socha et al. [8] on course timetabling) and ANTCOL-ET which is a modified version of ANTCOL (originally used by Costa and Hertz [9] to solve graph colouring problems). Both ant algorithms were hybridized with a hill climber and found that the simple ant system, ANTCOL, outperformed the more complex Max-Min algorithm. It was concluded that the performance of ant systems can be improved by adjusting the algorithm parameters.

Meta-heuristic approaches have also been used to improve solutions. Examples include tabu search [13,14,27], simulated annealing [10,15], genetic algorithms [16], memetic algorithms [6,17] great deluge algorithms [18], ant Colony [19], particle swarm optimization [20] and hybridizations of different techniques [11,12,24,25].

There are a number of survey papers on examination timetabling, which is essential reading for those new to the area [21-23].

In this work, we propose the Roulette Wheel Graph colouring heuristic (RWG) to construct examination timetables. We utilise the Largest Degree First heuristic to compute the difficulty of exams to be scheduled. Then, the degree of exams in conflict is used to determine the size of a segment in a *roulette wheel*. Finally, a roulette wheel selection mechanism is used select the next exam to be scheduled, in order to generate a feasible timetable.

The aim of this work is to investigate the capability of using a probability selection mechanism to construct examination timetables. The overall idea is to overcome the problem of repeatedly choosing the same sequence of exams to be scheduled, which is a common problem when using a deterministic graph colouring heuristic. In order to demonstrate the efficiency of the proposed heuristic, test it on the un-capacitated Carter's benchmark examination timetable dataset [28] (variant *b*, type I, see [22]) using the standard evaluation function given in [28].

2 Roulette Wheel Graph Colouring

In 1987 Baker [29] presented a simple selection schema called roulette wheel selection. This is a stochastic algorithm and is described as follows:

1. The individuals are mapped to contiguous segments of a line, such that each individual's segment is equal in size to its fitness.
2. A random number is generated and the individual whose segment spans the random number is selected.
3. The process is repeated until the desired number of individuals is obtained.

This technique is analogous to a roulette wheel with each segment proportional to its fitness. However, in this work, we compute the segment area by using Eq. 1:

$$S(i) = S(i-1) + d(i) / \sum_{i=1}^{n} d(i) \quad \text{for all } i \in \{1 \ldots n\},\ S(i-1) = 0 \text{ if } i = 1 \quad (1)$$

Where *S(i)* is the segment area and *d(i)* is the number of exams in conflict. To see how roulette wheel selection mechanism works, we give the following example. Suppose we have the exams as shown in Table 1, the number of exams in conflict is shown in column 3.

Table 1. An illustrative example of roulette wheel selection

Exam No	*Exams*	*Number of exams in conflicts*	Segment *Size*	Segment *Area (using Eq. 1)*
1	A	10	0.13	0.00-0.13
2	B	9	0.12	0.14-0.25
3	C	20	0.27	0.26-0.53
4	D	15	0.20	0.54-0.73
5	E	18	0.25	0.74-1.00

Then, the area of Segment can be calculated by using Eq. 1. For example, the Segment size for exams A and B in Table 1 are calculated as follows:

d(1)=10, d (2)=9 and $\sum_{i=1}^{n} d = 72$

so the sector size for A and B will be S(1) = S(1-1) + 10/72 = 0 + 0.13 = 0.13; S(2) = S(2-1) + 9/72 = 0.13 + 0.12 = 0.25, respectively.

Indeed, based on Table 1, exam C is the fittest one and occupies the largest interval (biggest segment); whereas exam number B is the least fit exam and has the smallest interval on the line. Based on the sector size in calculated in Table 1, Figure 1 shows the exams sector order.

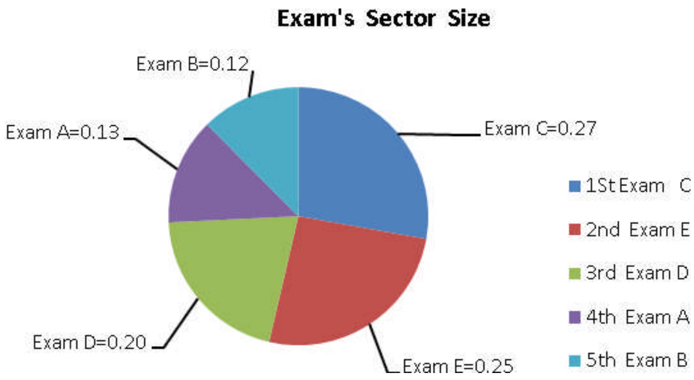

Fig. 1. Exams sector calculated using Eq. 1 for the example in Table 1

The pseudo-code of the Roulette Wheel Graph colouring heuristic is presented in Figure 2.

```
Step 1: Initialization
    - Calculate the degree of difficulty for each unscheduled
      exam e in a non-increasing order of the number of
      conflicts they have with other exams in unscheduled exam
      list E
    - Calculate the maximum span for each unscheduled exam e
      by using Eq. (1).
 Step 2: schedule exams into timeslots
   While (E≠Ø)
     {
      - Generate a random number r between [0, 1].
      - Select the exam e where r falls within its segment
        span.
      - Calculate the number of available clash free timeslots
        for the chosen exam e.
      - If number of available clash free timeslots > 0 then
          {
             • Schedule e to the minimum penalty period.
             • Remove e from Unscheduled list E.
          }
      - Else Break; // Terminate the procedure.

      } // end of while E≠Ø
 If E≠Ø then return the solution
```

Fig. 2. Pseudo-code of the Roulette Wheel Graph colouring heuristic.

In the initialization step, all exams in E are sorted on a decreasing order of the number of conflict they have with other exams. Then, we calculate the segment size for all exams based on Eq. 1. In the second step, we generate a random number *r* between [0, 1]. The exam whose segment span the random number is selected. Next, we calculate the number of available clash free timeslots for the selected exam *e*. If the number of available clash frees timeslots is greater than zero, allocate exam *e* to the minimum penalty clash free timeslot and remove exam *e* from unscheduled list *E*. If there is more than one timeslot that has the same minimum penalty, the timeslot is randomly selected from the set of minimum cost timeslots. If there is no available clash free timeslots for the current exam terminate the procedure. That is, the procedure returns an infeasible solution.

3 Results on Benchmark Examination Timetabling Dataset

The Roulette Wheel Graph colouring heuristic was tested on Carter's un-capacitated examination timetabling benchmark datasets [28] (variant *a*, type I, [22]) which contains 13 problem instances. These datasets have been used by many researchers in the literature since 1996 [22]. Table 2 shows the number of timeslots, number of exams and the number of student enrolments for these datasets.

Table 2. Un-capacitated standard Carter benchmark exam timetabling dataset

Data sets	Number of timeslots	Number of examinations	Number of Students
Car-f-92	32	543	18419
Car-s-91	35	682	16925
Ear-f-83	24	190	1125
Hec-s-92	18	81	2823
Kfu-s-93	20	461	5349
Lse-f-91	18	381	2726
Pur-s-93	43	2419	30032
Rye-s-93	23	486	11483
Sta-f-83	13	139	611
Tre-s-92	23	261	4360
Uta-s-92	35	622	21267
Ute-s-92	10	184	2750
Yor-f-83	21	181	941

The results (out of 20 runs for each instance) are shown in Table 3, comparing the results against other constructive heuristics. Results indicate that this simple heuristic is able to produce feasible initial solutions for all instances. Note that all the other algorithms being compared in Table 3 employed more advanced techniques i.e. fuzzy techniques, neural networks, ant algorithms (constructive and improvement heuristics) and tabu search hyper-heuristic. Our constructive heuristic is simple and generally applicable to produce feasible solutions for all 13 instances being studied.

It is known from the literature that for the benchmark dataset tested here, by employing the mostly used graph colouring heuristics such as largest degree and saturation degree, etc, alone cannot obtain feasible solutions for some of the difficult instances in Table 2. Other intelligent mechanisms need to be employed, or the graph colouring heuristics are repeated to obtain feasible solutions for advanced meta-heuristics [22]. Our new graph colouring heuristic with roulette wheel selection presents an effective and simple construction heuristic, and may be subsequently be improved by using a wide range of meta-heuristics.

Table 3. Results obtained from Roulette wheel Graph Colouring compared to constructive heuristics in the literature

Data sets	Our Results			Asmuni et al. [3]	Corr et al. [5]	Eley [4]	Burke et al. [2]
	Best	**% feasible**	**Standard deviation**				
Car-f-92	6.21	80	0.26	4.56	6.24	4.8	**4.53**
Car-s-91	7.01	90	0.11	**5.29**	7.21	5.7	5.36
Ear-f-83	42.81	85	0.35	37.02	49.44	**36.8**	37.92
Hec-s-92	12.90	70	0.08	11.87	13.57	**11.3**	12.25
Kfu-s-93	18.47	90	0.17	15.81	19.9	**15.0**	15.2
Lse-f-91	15.62	95	0.16	12.09	14.99	12.1	**11.33**
Pur-s-93	7.92	30	0.34	-	-	5.4	-
Rye-s-93	10.5	60	0.29	10.35	-	**10.2**	-
Sta-f-83	161.00	100	0.14	160.42	159.28	**157.2**	158.19
Tre-s-92	10.98	100	0.31	8.67	10.77	**8.8**	8.92
Uta-s-92	4.76	85	0.10	3.57	4.48	3.8	**2.88**
Ute-s-92	29.69	80	0.50	27.78	31.25	**27.7**	28.01
Yor-f-83	39.83	100	0.35	40.66	-	**39.6**	41.37

4 Conclusion

We have proposed a Roulette wheel Graph colouring heuristic for solving examination timetabling problems. We first utilize the Largest Degree First graph colouring heuristic to order exams. Then, based on the degree of exams in conflict, we determine the size of its segment in a roulette wheel. An exam is then selected if a generated random number falls within the exam segment. This overcomes the problem of repeatedly choosing the same sequence of exams to be scheduled. The current constructive heuristics shown in the literature quite often lead to low quality timetables or infeasible timetable especially for the more difficult problem instances in the benchmark dataset considered here. Our results show that the Roulette wheel Graph colouring heuristic is capable of producing feasible solutions for all problem instances.

References

1. Carter, M.W., Laporte, G.: Recent developments in practical examination timetabling. In: Burke, E.K., Ross, P. (eds.) PATAT 1995. LNCS, vol. 1153, pp. 3–21. Springer, Heidelberg (1996).
2. Burke, E.K., McCollum, B., Meisels, A., Petrovic, S., Qu, R.: A Graph-Based Hyper-Heuristic for Educational Timetabling Problems. European Journal of Operational Research 176, 177–192 (2007).
3. Asmuni, H., Burke, E.K., Garibaldi, J.M., McCollum, B.: Fuzzy multiple ordering criteria for examination timetabling. In: Burke, E.K., Trick, M.A. (eds.) PATAT 2004. LNCS, vol. 3616, pp. 334–353. Springer, Heidelberg (2005).
4. Eley, M.: Ant algorithms for the exam timetabling problem. In: Burke, E.K., Rudova, H. (eds.) PATAT 2007. LNCS, vol. 3867, pp. 364-382. Springer, Heidelberg (2007).
5. Corr, P.H., McCollum, B., McGreevy, M.A.J., McMullan, P.: A New Neural Network Based Construction Heuristic for the Examination Timetabling Problem. In: The International Conference on Parallel Problem Solving From Nature (PPSN IX). LNCS, vol. 4193, pp. 392-401. Springer, Heidelberg (2006).
6. Burke, E.K., Newall, J.P., Weare, R.F.: A memetic algorithm for university exam timetabling. In: Burke, E.K., Ross, P. (eds.) PATAT 1995. LNCS, vol. 1153, pp. 241–250. Springer, Heidelberg (1996).
7. de Werra, D.: An introduction to timetabling. European Journal of Operational Research, 19,151-162 (1985).
8. Socha, K., Sampels, M., Manfrin, M.: Ant algorithms for the university course timetabling problem with regard to state-of-the-art. In: Proceedings of the 3rd European Workshop on Evolutionary Computation in Combinatorial Optimisation, pp. 334-345. Essex, UK, April (2003).
9. Costa, D., Hertz, A.: Ant can colour graphs. Journal of Operational Research Society, 48, 295-305, (1997).
10. Burke, E.K., Newall, J.: Enhancing timetable solutions with local search methods. In: Burke, E.K., De Causmaecker, P. (eds.) PATAT 2002. LNCS, vol. 2740, pp. 195–206. Springer, Heidelberg (2003).
11. Caramia, M., Dell'Olmo, P., Italiano, G. F.: New algorithms for examination timetabling. In: Naher, S., Wagner, D. (eds.) WAE 2000. LNCS, vol. 1982, pp. 230-241. Springer, Heidelberg (2001).
12. Merlot, L.T.G., Borland, N., Hughes, B.D., Stuckey, P.J.: A hybrid algorithm for the examination timetabling problem. In: Burke, E.K., De Causmaecker, P. (eds.) PATAT 2002. LNCS, vol. 2740, pp. 207–231. Springer, Heidelberg (2003).
13. Di Gaspero, L., Schaerf, A.: Tabu search techniques for examination timetabling. In: Burke, E.K., Erben, W. (eds.) PATAT 2000. LNCS, vol. 2079, pp. 104–117. Springer, Heidelberg (2001).
14. Kendall, G. and Hussin, N.M.: Tabu Search Hyper-Heuristic Approach to the Examination Timetabling Problem at University Technology MARA. In: Burke, E.K., Trick, M. (eds.) PATAT 2004. LNCS, vol. 3616, pp. 199-218. Springer, Heidelberg (2005).
15. Thompson, J., Dowsland, K.: A robust simulated annealing based examination timetabling system. Computers Operations Research 25, 637–648 (1998).
16. Burke, E.K., Elliman, D.G., Weare, R.F.: A hybrid genetic algorithm for highly constrained timetabling problems. In: Proceedings of the 6th International Conference on Genetic Algorithms (ICGA'95), pp. 605-610, Morgan Kaufmann, San Francisco, CA, USA (1995).
17. Burke, E.K., Landa Silva, J.D.: The design of memetic algorithms for scheduling and timetabling problems. In: Hart, W. E., Krasnogor, N., Smith, J. E. (eds.) Studies in

fuzziness and soft computing: Recent advances in memetic algorithms and related search technologies, vol. 166, pp. 289–312. Springer, Heidelberg (2004).

18. Burke, E.K., Bykov, Y., Newall, J.P., Petrovic, S.: A time-predefined local search approach to exam timetabling problems. IIE Transactions on Operations Engineering 36, 509–528 (2004).
19. Dowsland, K., Thompson, J.: Ant colony optimisation for the examination scheduling problem. Journal of the Operational Research Society 56, 426–439, (2005).
20. Chu, S.C., Chen, Y.T., Ho, J.H.: Timetable scheduling using particle swarm optimization. In: First international conference on innovation computing, Information and control. vol. 3, pp. 324-327. IEEE Computer Society, Washington, DC, USA (2006).
21. Burke, E.K., Kingston, J., de Werra, D.: In: Gross, J., Yellen, J. (eds.) Applications to timetabling. Handbook of Graph Theory, pp. 445–474. Chapman Hall/CRC Press (2004).
22. Qu, R., Burke, E.K., McCollum, B., Merlot, L.T.G., Lee, S.Y.: A Survey of Search Methodologies and Automated System Development for Examination Timetabling. Journal of Scheduling, 12(1): 55-89, DOI 10.1007/s10951-008-0077-5 (2009).
23. Burke, E.K., Petrovic, S.: Recent research directions in automated timetabling. European Journal of Operational Research 140, 266–280 (2002).
24. Abdullah, S., Burke, E.K.: A Multi-start Large Neighbourhood Search Approach with Local Search Methods for Examination Timetabling. In Long, D., Smith, S.F., Borrajo, D., McCluskey, L.(eds.) ICAPS 2006, pp. 334-337, Cumbria, UK (2006).
25. Burke, E.K., Eckersley, A.J., McCollum, B., Petrovic S., Qu, R.: Hybrid variable neighbourhood approaches to university exam timetabling. Accepted by European Journal of Operational Research, Technical Report NOTTCS-TR-2006-2, School of Computer Science, University of Nottingham, United Kingdom. (2009).
26. Ayob, M., Malik, A.M.A., Abdullah, S., Hamdan, A.R., Kendall, G., Qu, R.: Solving a Practical Examination Timetabling Problem: A Case Study. In: Gervasi, O., Gavrilova, M. (Eds.) ICCSA 2007, LNCS, vol. 4707, pp. 611–624. Part III. Springer, Heidelberg (2007).
27. White, G. M., Xie, B. S.: Examination timetables and tabu search with longer-term memory. In Burke, E.K., Erben, W. (eds.) PATAT 2000, LNCS, vol. 2079, pp. 85–103. Springer, Heidelberg (2001).
28. Carter, M.W., Laporte, G., Lee, S.Y.: Examination timetabling: Algorithmic strategies and applications. Journal of Operational Research Society 47(3), 373–383 (1996).
29. Baker, J.: Reducing bias and inefficiency in the selection algorithm. In Proceeding of the Second International Conference on Genetic Algorithms (ICGA'87), pp. 14–21, L. Erlbaum Associates Inc. Hillsdale, NJ, USA (1987).