Solving Examination Timetabling Problem Using Partial Exam Assignment with Hill Climbing Search

Ashis Kumar Mandal, M N M Kahar,
Faculty of Computer Systems & Software Engineering,
University Malaysia Pahang, Lebuhraya Tun Razak,
Gambang, 26300 Kuantan, Pahang
ashis@hstu.ac.bd, mnizam@ump.edu.my

***Abstract* — This paper describes a method that combines graph heuristics and hill climbing for addressing the examination timetable problem. In this approach, all exams are ordered with graph heuristic ordering approach and partial exams are considered for scheduling. These partial scheduled exams are then improved using hill climbing until all exams have been successfully scheduled. Various *exam assignment values* with different graph heuristics ordering have been investigated. The proposed approach has been tested over the twelve Toronto benchmark datasets. The experimental results and comparison with other methods demonstrate that the proposed approach is able to produce good quality timetable.**

Keywords - Examination timetabling, scheduling, graph heuristics, hill-climbing, Toronto benchmark datasets

I. Introduction

Examination timetabling is a common problem faced by all academic institutions when scheduling exams. Like most timetabling problems, generating good quality examination timetables is a challenging and time consuming task due to its combinatorial and highly constrained nature. Hence, to generate better quality solution, numerous approaches have been proposed in the literature. In [1], they provide a good survey on search methodologies and automated approaches for solving examination timetabling problem. This survey demonstrates that many approaches such as graph heuristic [1], tabu search [2], simulated annealing [3], great deluge[4], late acceptance hill climbing [5], evolutionary algorithms [6-8], constraint programming [9], case based reasoning [10] and fuzzy methodologies [11] have been successfully being applied to solve the examination timetabling problem.

In this work, we formulate the examination timetabling problem based on partial exams construction and improvement strategy. This is done by combining method of graph heuristics and hill climbing strategy. Firstly, the exams are ordered using graph heuristic and these ordered exams are scheduled based on the *exam assignment value*. Next, the partially scheduled exams are improved using hill climbing method. The remaining exams are then scheduled and improved as in the above process until all exams have been successfully scheduled. We experiment with different graph heuristic sorting strategies (e.g. LD, LE, LWD and SD) and *exams assignments value*. This proposed approach was applied on the Toronto benchmark datasets. The experimental results clearly show that this method is able to produce good quality solution.

The rest of the paper is organized as follows. In Section II, we discuss related theory of the benchmark dataset associated with examination timetable problem. Our approach of partial graph heuristic with hill climbing is presented in section III. In section IV, we describe the experimental setup taken in solving the problem. Section V compares the obtained results with other results reported in the literature. Finally, the conclusion and future direction is presented in section VI.

II. Related Theory

A. *Examination timetable problem*

Examination timetabling is a scheduling problem where exams are allocated into a limited number of timeslots subject to a set of constraints [12]. Every examination timetabling problem has specific hard constraints and some soft constraints. Hard constraints are those constraints that must be satisfied in order for the exam timetable to be acceptable (also known as feasible solution), while the soft constraints need to be satisfied as much as possible, but it can be violated if necessary with a penalty value. Example of the hard and soft constraint can be seen in [1]. Although the hard and soft constraints differ considerably from one institution to another, all timetabling systems will never violate the hard constraints and tries to minimize the value of the soft constraints. Therefore, the soft constraints are used to determine the quality of the timetable.

B. *The Toronto Benchmark*

The most widely used examination benchmark dataset was introduced by [13]. This dataset is also known as the Toronto benchmark dataset. The original version data files are available at http://www.asap.cs.nott.ac.uk/resources/data.shtml. This dataset is an un-capacitated examination timetabling benchmark dataset where it assumes an unlimited number of seats during exam assignment. The Toronto dataset consists of 13 problem instances. Table I shows the details information of the dataset. The Toronto examination timetable hard constraint insists that no students are allowed to seat two or more exams simultaneously (also known as clashing constraint). In addition, the soft constraint is to spread the exam evenly so that students get ample time for last minute preparation before the next exams. The objective function is shown in eq.1 [28]. Based on eq.1, penalty value 32 is given for assigning exams consecutively; penalty value 16 with a timeslot gap in between exams follows with a penalty value 8, 4 and 2 for 2, 3 and 4 timeslot gap between exams respectively.

TABLE I. TORONTO DATASET

Dataset	Number of timeslots	Number of Exams	Number of Students	Conflict Density
1. car-s-91 -I	35	682	16925	0.13
2. car-f-92-I	32	543	18419	0.14
3. ear-f-83 -I	24	190	1125	0.27
4. hec-s-92 -I	18	81	2823	0.42
5. kfu-s-93	20	461	5349	0.06
6. lse-f-91	18	381	2726	0.06
7. pur-s-93 -I	42	2419	30029	0.03
8. rye-s-93	23	486	11483	0.07
9. sta-f-83-I	13	139	611	0.14
10. tre-s-92	23	261	4360	0.18
11. uta-s-92 -I	35	622	21267	0.13
12. ute-s-92	10	184	2750	0.08
13. yor-f-83 -I	21	181	941	0.29

$$\min f(X) = \frac{\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} c_{ij} \times proximity\ (t_i,t_j)}{M} \qquad \text{(eq.1)}$$

Where

$$proximity(t_i, t_j) = \begin{cases} \dfrac{2^5}{2^{|t_i - t_j|}}\ if\ 1 \le |t_i - t_j| \le 5 \\ 0 \qquad\qquad Otherwise \end{cases}$$

N is the number of examinations
X is complete timetable solution
M is the total number of students
T is the number of available time slots.
c_{ij} is the conflict matrix, where each element in the matrix is the number of students taking examination i and j, and where $i, j \in \{1,..., N\}$
t_k $(1 \le t_k \le T)$ specifies the assigned timeslot for examination k ($k \in \{1,..., N\}$)

The main objective of solving the Toronto dataset problem is to satisfy the hard constraint and minimize the penalty value of the soft constraint as given in eq.1.

C. Initial solution using Graph Colouring Heuristics

The examination timetable problem can be represented as graph colouring problem where the examinations represent vertices and the hard constraints (i.e. conflict between exams) are shown by the edges between the vertices. Here adjacent vertices have different colour and colour of the vertices indicate different timeslots in the timetable. Graph heuristics is based on ordering strategy where the most difficult exams are chosen for scheduling first. The difficult exams are measured with various graph heuristics. The most commonly used graph colouring heuristics ordering strategy seen in the literature includes largest degree (LD), largest enrolment (LE), largest weighted degree (LWD) and saturation degree (SD). Description of each ordering strategies are as follows:

- Largest degree (LD): This technique orders the exams based on the largest number of conflicting examinations.
- Largest weighted degree (LWD): This heuristic is similar to the largest degree except the exams are ordered based on the number of students in conflict.
- Largest enrolment (LE): The exams are ordered based on the number of registered students in the exams.
- Saturation degree (SD): The exams are ordered based on the number of the remaining time slots available, exams with the least number of available timeslots in the timetable are given priority to be scheduled first. SD is a dynamic heuristic where the ordering of exams is updated as the exams being scheduled.

Graph heuristics are being used widely to produce an initial solution for the examination timetable. In[14], they investigated performance of these four heuristic ordering strategies to generate the initial solution. Another approach is using adaptive heuristic technique [15] where exams are ordered primarily with a particular heuristic and later altered according to adaptive strategy. Similarly, fuzzy strategy of selecting heuristic and ordering was also implemented in timetabling problem[11]. Some researchers also used graph heuristic hybridizations to generate the initial solution [16-17]. Recently, in[18], they used graph heuristic to solve real world university exam timetabling problem which produced better quality solution than university's existing software.

D. Improvement Method

The improvement method concentrates on minimizing the soft constraint value (i.e. violation of the penalty value). Many improvement methods can be seen in the literature. [5] introduced late-acceptance hill climbing (LAHC) that is based on intelligent memory management to solve the timetabling problems. Similarly, tabu search also have been used to solve the problem by accepting worse moves to escape from local optima and tabu list are used as memory for storing such information [18]. To encourage exploration of the search area, simulated annealing has been applied to accept worse solution using certain probability function with temperature parameter [19]. Another approach similar to simulated annealing is the great deluge algorithm where acceptance of solution is determined by certain boundary value[19] that is decreased over time during the searching. The advantages of this algorithm are that it requires less parameter tuning and have capabilities to produce better result than simulated Annealing in many cases[12].

Genetic Algorithm is also popular which is based on evolutionary strategy. Initial solutions are evolved with some steps that involve selection, recombination, and mutation, and different solutions are produced with good quality[8]. Besides, other population based methods include ant colony optimization[20], particle swam optimization[21], bee colony algorithm[7] have been used successfully for improving the examination timetabling solutions. Recently, hybridization of two or more techniques has been proposed in order to better exploit and explore the search space to yield a better solution. Examples is the hybridization of ant colony algorithm with tabu search[22], artificial bee colony algorithm with late acceptance hill climbing[23], variable neighborhood search with genetic algorithm [24] and many others.

Hill climbing approach is the preferred local search methods because of its simplicity in implementation of the algorithm. In hill climbing, a candidate solution is accepted providing it has better or equivalent cost than the current one. Main merits of this algorithm are that it does not require any parameter setting and produce solution quickly albeit tendency of trapping in local optima. Recently, many researchers combine hill climbing with other methods in order to get around its limitation. For example, in [25], they hybridize hill climbing with constraint programming and simulated annealing. Their approaches were able to produce good quality solution when tested on some benchmark datasets. In [26], hyper-heuristic with hill climbing was also tested for examination timetable problem. The simplicity of hill-climbing attracted us to further explore this method.

III. Proposed approach

The process of producing examination timetable include, firstly generating the initial solution which aim to satisfy the hard constraint only regardless of the soft constraint value. This step normally referred to as constructive heuristics which normally involve graph heuristics. Next, the initial solutions are improved aiming to minimize the soft constraint value while maintaining the feasibility of the timetable. The improvement steps involve method such as hill climbing [12], simulated annealing [27], great deluge algorithm[4], genetic algorithm[8]and many others. Usually, it is observed that initial solution biases the result of improvement steps, where good initial solution would produce good improved solution [4].Additionally, if the initial solution is bad (i.e. local optima), sometimes the improvement technique unable to work effectively in producing quality solution[16]. Therefore, we propose a partial graph heuristics with hill climbing to solve the examination timetabling problem. Following paragraphs and Fig.1 describe how we utilize partial graph heuristics and hill climbing to solve the examination timetabling problem.

Initially, all examinations are ordered according to a particular graph heuristic ordering approach. We implemented SD approach (with initial ordering using LE, LD and LWD) because SD is able to produce better result compared to other ordering approaches [15]. Afterwards, partial exams are selected from total exams derived from the *exam assignment value*. *Exam assignment value* indicates the number of exams to be scheduled and we experiment with different number of exams (we refer to as partial schedule exam). These exams are then scheduled to the timeslots whilst satisfying the hard constraints. In the next step, hill climbing (HC) is used to improve the partial scheduled exam with the aim of satisfying the soft constraint (i.e. minimizing the penalty value). The above process repeats for the next batch of partial exams to be scheduled and these scheduled exams are then improved using hill climbing method. The entire process repeats until all exams have been scheduled.

```
Step1. Choose heuristics and do initial ordering based on
       [SD(LE),SD(LWD),SD(LE)]
Step2. Set examination assignment value
Step 3.
       while until end of all exam assigned to timeslots
          schedule partially exams
          calculate (temporary) penalty cost
          use hill climbing(HC) for improvement
       end while
Step4. calculate final penalty cost
```

Fig.1 Partial graph heuristics with hill climbing algorithm

IV. Experimental setup

In this section, we describe the experimental setup to allow reproducibility. The algorithm was evaluated on Toronto datasets shown in Table I. The programs were implemented in Java and performed on Intel Core2Duo (3 GHz) PCs running Windows 7 Professional SP3, with a Java 1.7 JRE. We experiment with four different *exam assignment values* which are 10%, 25%, 50% and 75% from the total number of exams. For example, if the total number of exams are 100, *exam assignment value* of 10% will involve scheduling 10 exams from the list (10% x 100 exams = 10 exams) and so on until all exams have been scheduled. Same goes to other value of percentage for the *exam assignment value.*

For the improvement of solution quality with hill climbing (HC), Initially 50,000 iterations was set as stopping criteria, secondary stopping criteria was also set when no improvement in 10,000 iterations. We run the experiment 30 times for each case. During the improvement phase, we consider three neighborhood structures as follows:

i) Moving - randomly select an exam and move the selected exam to a randomly selected timeslot.
ii) Swapping - randomly select two exams and swap their timeslots.
iii) Swapping timeslot -select two timeslots randomly and move all exams between the two timeslots.

The above three (3) neighbourhood structures are used during the improvement phase. However, we only accept neighbourhood structures that give an improvement on the penalty value in each iteration.

V. Results And Discussion

In this section, we discuss on the result obtained from the experiments. Table II shows the result implemented on Toronto dataset using *traditional graph heuristics (GH) with hill climbing (HC)* compared to *partial graph heuristics(GH) with hill climbing(HC)*.The table summarizes the obtained best cost and the mean cost for all datasets. For the traditional graph heuristics with hill climbing, we did experiments using six (6) graph heuristic ordering strategies (i.e. LD, LWD, SE, SD(LD), SD (LWD), SD (LE)). After running the experiments for 30 times, we keep the best solution as initial solution. This solution is then improved using hill climbing.

TABLE II. COMPARISONS OF RESULT BETWEEN THE TRADITIONAL GH WITH HC AND PARTIAL GH WITH HC

Datasets	**Traditional graph heuristic with hill climbing**			**Partial graph heuristics with hill climbing**				**Percentage of improvement %**
	Constructive with Graph Heuristic Ordering	**Hill Climbing**		**Hill Climbing**		**Graph Heuristic ordering**	**Exam Assignment value (%)**	
		Best	**Mean**	**Best**	**Mean**			
car-s-91	8.33 – LD	5.54	5.74	5.08	5.27	SD(LWD)	10%	8.30
car-f-92	7.00 – LD	4.72	4.92	4.23	4.48	SD(LE)	10%	10.38
ear-f-83	52.35 - SD(LE)	40.48	41.68	37.06	40.13	SD(LE)	10%	8.45
hec-s-92	16.21 - SD(LWD)	12.06	12.48	11.19	11.81	SD(LWD)	10%	7.21
kfu-s-93	23.68 - (LD)	16.63	16.77	14.53	15.69	SD(LE)	10%	12.63
lse-f-91	18.83 - (LE)	12.29	13.03	11.21	11.9	SD(LD)	10%	8.79
rye-s-93	18.28 - SD(LD)	10.38	10.8	9.45	9.89	SD(LE)	10%	8.96
sta-f-83	166.43 - SD(LE)	157.54	158.4	157.23	158.22	SD(LE)	10%	0.20
tre-s-92	12.07- SD(LE)	9.54	9.82	8.59	8.87	SD(LWD)	10%	9.96
uta-s-92	5.53 – LE	3.86	3.98	3.43	3.58	SD(LE)	10%	11.14
ute-s-92	38.03 – SD(LD)	28.94	30.15	26.57	28.39	SD(LWD)	10%	8.19
yor-f-83	49.8 – LD	40.02	41.27	38.83	41.2	SD(LE)	10%	2.97

TABLE III. BEST RESULT OBTAIND BY PROPOSED APPROACH COMPARED TO THE BEST RESULT IN THE LITERATURES ON TORONTO DATASET

Datasets	**Carter et al. [28]**	**Rahman et al.[29]**	**Turabieh and Abdullah [30]**	**Bruke et al. [24]**	**Caramia et al.[31]**	**Pillay and Banzhaf [32]**	**Sabar et al. [17]**	**Our approach**	**Comparison with the best result (%)**
car-s-91	7.10	5.12	4.8	**4.6**	6.6	4.97	5.14	5.08	9.45
car-f-92	6.20	4.41	4.1	**3.9**	6.0	4.28	4.70	4.23	7.80
ear-f-83	36.40	36.91	34.92	32.8	**29.3**	35.86	37.86	37.06	20.94
hec-s-92	10.80	11.31	10.73	10.0	**9.2**	11.85	11.90	11.19	17.78
kfu-s-93	14.00	14.75	**13.0**	**13.0**	13.8	14.62	15.30	14.53	10.53
lse-f-91	10.5	11.41	10.01	10.0	**9.6**	11.14	12.33	11.21	14.36
pur-s-93	3.9	5.87	4.73	–	**3.7**	4.73	5.37	-	-
rye-s-93	7.3	9.61	9.65	–	**6.8**	9.65	10.71	9.45	28.04
sta-f-83	161.5	157.52	158.26	**156.9**	158.2	158.33	160.12	157.23	0.21
tre-s-92	9.6	8.76	**7.88**	7.9	9.4	8.48	8.32	8.59	8.27
uta-s-92	3.5	3.54	**3.2**	**3.2**	3.5	3.40	3.88	3.43	6.71
ute-s-92	25.8	26.25	26.11	24.8	**24.4**	28.88	32.67	26.57	8.17
yor-f-83	41.7	39.67	36.22	**34.9**	36.2	40.74	40.53	38.83	10.12

We compare the solution qualities of the proposed approach (*partial GH with HC*) with other approaches using following calculation

$$improvement\ (\%) = \frac{Our\ solution - Other\ solution}{Other\ solution} x100\% \quad \text{(eq2)}$$

In Table II, the partial GH with HC approach clearly outperforms the traditional approach for all of the Toronto dataset. Overall, the partial GH with HC is able to produce solution that is around 8% of improvement compared to the traditional GH with HC. The highest percentage of improvement is obtained for kfu-s-93 dataset with 12.63% (where 14.53 for partial GH with HC compared to 16.63 for traditional GH with HC). The lowest percentage of improvement is obtained for sta-f-83 dataset with only 0.2% (where 157.23 for partial GH with HC compared to 157.54 for traditional GH with HC). It is noted that these comparative results are observed while taking consideration of the best solutions. With considering average values in comparison, similar result can be found.

Additionally, for each instance the best solution is produced for partial GH with HC when SD is employed as the graph heuristic ordering approach (see Table II). SD is able to work better in this proposed approach compare to LE, LD and LWD because of its dynamic nature that sort exams based on their difficulty to be scheduled in the timeslots. Besides that, the best solution were produced when using small *exam assignment value* which is 10% compared to using 25%, 50% and 75%. When small *exam assignment value* is used, the scheduled exams undergo several iterations of improvement, thus reducing the objective function.

Next, we compare our proposed technique with other state-of-the-art techniques reported in the scientific literature (see Table III). We are able to produce solution for all of the datasets except for pur-s-93 dataset. Although our proposed approach did not produce the best result for any of the datasets, its performance is comparable with other results in Table III. In comparison with the best reported results, our best results differ in average around 10% only and for sta-f-83 dataset, we are able to produce results close to the best reported results [17] [18]. Even though our approach is unable to outperform other reported result, the advantage of our proposed algorithm is that it is simple to implement and does not require complex parameter setting like other algorithms in the Table III.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented combination of different graph heuristics with hill climbing approach to solve the examination timetabling problem. Our approach is different from other approaches as we introduced partial examination assignment concept. We evaluated our algorithm using the well-known Toronto benchmark datasets. Results showed that the method produced better results than the traditional graph heuristic with the hill climbing approach, and competitive result compared to others reported results. Furthermore, different examination assignment value was analyzed on solution generation and improvement. It is shown that small exam assignment value produces better solutions.

In our future works, focus will be given to further improve the proposed algorithm by including *improvement cycle*. We are optimistic that this will produce better result. We are also motivated to apply other meta-heuristic approaches, such as great deluge algorithm and simulated annealing to replace the hill climbing. Finally, we intend to apply the algorithm on International Timetabling Competition datasets (ITC2007).

REFERENCES

[1] B. Hussin, A. S. H. Basari, A. S. Shibghatullah, S. A. Asmai, and N. S. Othman, "Exam timetabling using graph colouring approach," in *Open Systems (ICOS), 2011 IEEE Conference on*, 2011, pp. 133-138.

[2] L. Di Gaspero and A. Schaerf, "Tabu Search Techniques for Examination Timetabling," in *Practice and Theory of Automated Timetabling III*. vol. 2079, E. Burke and W. Erben, Eds., ed: Springer Berlin Heidelberg, 2001, pp. 104-117.

[3] J. M. Thompson and K. A. Dowsland, "A robust simulated annealing based examination timetabling system," *Comput. Oper. Res.,* vol. 25, pp. 637-648, 1998.

[4] M. N. M Kahar and G. Kendall, "A great deluge algorithm for a real-world examination timetabling problem," *Journal of the Operational Research Society,* 2013.

[5] E. K. Burke and Y. Bykov, "The late acceptance hill-climbing heuristic," Technical report CSM-192 Computing Science and Mathematics, University of Stirling, UK2012.

[6] E. K. Burke and J. P. Newall, "A multistage evolutionary algorithm for the timetable problem," *Evolutionary Computation, IEEE Transactions on,* vol. 3, pp. 63-74, 1999.

[7] M. Alzaqebah and S. Abdullah, "Artificial bee colony search algorithm for examination timetabling problems," *International Journal of Physical Sciences,* vol. 6, pp. 4264-4272, 2011.

[8] S. Innet, "A noval approach of genetic algorithm for solving examination timetabling problems: A case study of Thai Universities," in *Communications and Information Technologies (ISCIT), 2013 13th International Symposium on*, 2013, pp. 233-237.

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

[10] E. K. Burke, S. Petrovic, and R. Qu, "Case-based heuristic selection for timetabling problems," *Journal of Scheduling,* vol. 9, pp. 115-132, Apr 2006.

[11] H. Asmuni, E. K. Burke, J. M. Garibaldi, B. McCollum, and A. J. Parkes, "An investigation of fuzzy multiple heuristic orderings in the construction of university examination timetables," *Computers & Operations Research,* vol. 36, pp. 981-1001, 4// 2009.

[12] R. Qu, E. K. Burke, B. McCollum, L. T. G. Merlot, and S. Y. Lee, "A survey of search methodologies and automated system development for examination timetabling," *Journal of Scheduling,* vol. 12, pp. 55-89, 2009/02/01 2009.

[13] M. W. Carter and G. Laporte, *Recent developments in practical examination timetabling*: Springer, 1996.

[14] S. Casey and J. Thompson, "GRASPing the examination scheduling problem," in *Practice and Theory of Automated Timetabling IV*, ed: Springer, 2003, pp. 232-244.

[15] E. Burke, R. Qu, and A. Soghier, "Adaptive selection of heuristics for improving exam timetables," *Annals of Operations Research,* pp. 1-17, 2012/06/26 2012.

[16] N. R. Sabar, M. Ayob, G. Kendall, and Q. Rong, "Grammatical Evolution Hyper-Heuristic for Combinatorial Optimization Problems," *Evolutionary Computation, IEEE Transactions on,* vol. 17, pp. 840-861, 2013.

[17] N. Sabar, M. Ayob, R. Qu, and G. Kendall, "A graph coloring constructive hyper-heuristic for examination timetabling problems," *Applied Intelligence,* vol. 37, pp. 1-11, 2012/07/01 2012.

[18] M. N. M. Kahar and G. Kendall, "The examination timetabling problem at Universiti Malaysia Pahang: Comparison of a constructive heuristic with an existing software solution," *European Journal of Operational Research,* vol. 207, pp. 557-565, Dec 1 2010.

[19] G. Dueck, "New optimization heuristics: the great deluge algorithm and the record-to-record travel," *Journal of Computational physics,* vol. 104, pp. 86-92, 1993.

[20] M. Eley, "Ant Algorithms for the Exam Timetabling Problem," in *Practice and Theory of Automated Timetabling VI*. vol. 3867, E. Burke and H. Rudová, Eds., ed: Springer Berlin Heidelberg, 2007, pp. 364-382.

[21] M. Alinia Ahandani, M. T. Vakil Baghmisheh, M. A. Badamchi Zadeh, and S. Ghaemi, "Hybrid particle swarm optimization transplanted into a hyper-heuristic structure for solving examination timetabling problem," *Swarm and Evolutionary Computation,* vol. 7, pp. 21-34, 12// 2012.

[22] Z. Naji Azimi, "Hybrid heuristics for Examination Timetabling problem," *Applied Mathematics and Computation,* vol. 163, pp. 705-733, 4/15/ 2005.

[23] M. Alzaqebah and S. Abdullah, "An adaptive artificial bee colony and late-acceptance hill-climbing algorithm for examination timetabling," *Journal of Scheduling,* vol. 17, pp. 249-262, 2014/06/01 2014.

[24] E. K. Burke, A. J. Eckersley, B. McCollum, S. Petrovic, and R. Qu, "Hybrid variable neighbourhood approaches to university exam timetabling," *European Journal of Operational Research,* vol. 206, pp. 46-53, 10/1/ 2010.

[25] L. T. Merlot, N. Boland, B. D. Hughes, and P. J. Stuckey, "A hybrid algorithm for the examination timetabling problem," in *Practice and theory of automated timetabling IV*, ed: Springer, 2003, pp. 207-231.

[26] G. Kendall and N. M. Hussin, "A tabu search hyper-heuristic approach to the examination timetabling problem at the MARA university of technology," in *Practice and Theory of Automated Timetabling V*, ed: Springer, 2005, pp. 270-293.

[27] D. Zhang, Y. Liu, R. M'Hallah, and S. C. Leung, "A simulated annealing with a new neighborhood structure based algorithm for high school timetabling problems," *European Journal of Operational Research,* vol. 203, pp. 550-558, 2010.

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

[29] S. Abdul Rahman, A. Bargiela, E. K. Burke, E. Özcan, B. McCollum, and P. McMullan, "Adaptive linear combination of heuristic orderings in constructing examination timetables," *European Journal of Operational Research,* vol. 232, pp. 287-297, 1/16/ 2014.

[30] H. Turabieh and S. Abdullah, "A Hybrid Fish Swarm Optimisation Algorithm for Solving Examination Timetabling Problems," in *Learning and Intelligent Optimization*. vol. 6683, C. C. Coello, Ed., ed: Springer Berlin Heidelberg, 2011, pp. 539-551.

[31] M. Caramia, P. Dell'Olmo, and G. F. Italiano, "Novel local-search-based approaches to university examination timetabling," *INFORMS Journal on Computing,* vol. 20, pp. 86-99, 2008.

[32] N. Pillay and W. Banzhaf, "A study of heuristic combinations for hyper-heuristic systems for the uncapacitated examination timetabling problem," *European Journal of Operational Research,* vol. 197, pp. 482-491, 2009.