Solving Examination Timetabling Problem using Partial Exam Assignment with Great Deluge Algorithm

Ashis kumar Mandal
Faculty of Computer Systems & Software Engineering
University Malaysia Pahang
Kuantan, Pahang
ashis@hstu.ac.bd

M N M Kahar
Faculty of Computer Systems & Software Engineering
University Malaysia Pahang
Kuantan, Pahang
mnizam@ump.edu.my

***Abstract*—Constructing a quality solution for the examination timetable problem is a difficult task. This paper presents a partial exam assignment approach with great deluge algorithm as the improvement mechanism in order to generate good quality timetable. In this approach, exams are ordered based on graph heuristics and only selected exams (partial exams) are scheduled first and then improved using great deluge algorithm. The entire process continues until all of the exams have been scheduled. We implement the proposed technique on the Toronto benchmark datasets. Experimental results indicate that in all problem instances, this proposed method outperforms traditional great deluge algorithm and when comparing with the state-of-the-art approaches, our approach produces competitive solution for all instances, with some cases outperform other reported result.**

***Keyword*—graph heuristic, great deluge algorithm, timetable**

I. INTRODUCTION

In the last few decades, the examination timetabling problem has been studied vastly in the operational research community due to its complexity and practical significance in education institutions. Most of the educational institutions face a challenging task of scheduling these examinations. The educational institution requires various requirements that need to be fulfilled in generating the timetable and these requirements are referred to as constraints. The constraints can be divided into hard and soft constraints.

In recent times, various techniques have been reported in the literature to solve the examination timetabling problem. One of the early approaches is the graph heuristic[1] which is popular for constructing the initial solution. The initial solution is then improved using local search meta-heuristic method. Examples of these improvement methods include tabu search[2], simulated annealing[3], great deluge[4], late acceptance hill climbing[5]. In population search based heuristics, more than one solution, called population, are considered at a time in generating an improved solution. Examples of these approaches include genetic algorithm[6], artificial bee colony algorithm[8], and scatter search algorithm[7] have been successfully applied to solve the examination timetabling problem. Recently, hybrid approach which combines more than one heuristics has been the major focus of many researchers in solving the examination timetabling problem. For instance, [8][9] hybridized artificial bee colony algorithm with late acceptance hill climbing in improving the solution quality. In [10], they hybridized tabu search with the memetic approach and produced some of the best results in Toronto exam datasets. Hyper heuristic is also a relatively new domain for solving examination timetabling problem. Hyper heuristic is a higher level heuristic that properly select lower level heuristic to solve the optimization problem. Graph colouring hyper heuristic [11], harmony search based hyper heuristic [12], Monte Carlo based hyper-heuristics [13] are some examples which have been successful in solving this timetabling problem.

In this work, we formulate the exam timetabling problem based on partial exams construction and improvement strategy. This is done by combining method of graph heuristics and great deluge algorithm. Firstly, exams are ordered using graph heuristic and partially selected exams are scheduled based on the *exam assignment value* (which indicates how many exams are selected for scheduling). Next, the partially scheduled exams are improved using great deluge algorithm. The remaining exams are then scheduled and improved as in 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 *exam assignments value*. This proposed approach is tested on the Toronto benchmark datasets. The experimental results clearly show that this method is able to provide a good quality solution.

The rest of the paper is organized as follows. In Section II, we discuss related background on examination timetable problem and benchmark datasets that being used as the test cases. In section III and IV, background on graph heuristic and great deluge algorithms have been highlighted respectively. Our approach of partial graph heuristic with great deluge algorithm is presented in section V. In section VI and VII, we describe the experimental setup and obtained result along with discussion on the result respectively. Finally, conclusions and future direction are presented in section VIII.

II. PROBLEM STATEMENT

Examination timetabling is a scheduling problem where exams are allocated into a limited number of timeslots subject to a set of constraints [14]. The problem contains hard and soft constraints. The hard constraints are those constraints that must be satisfied for the final exam timetable to be accepted (refer to as feasible solution). Soft constraints need to be satisfied as much as possible, but it can be violated if necessary. Most of the times, these soft constraints are defined by an objective function, which is used to determine the quality of the timetable produces. In this study, we investigate Toronto datasets. Details of the datasets as well as the objective function are described below:

A. The Toronto Benchmark

The Toronto dataset was introduced by [15] and were widely used as a test bed for the examination timetabling problem. It consists of 13 problems and can be downloaded at http://www.asap.cs.nott.ac.uk/resources/data.shtml. Table I shows details information of the datasets. The hard and soft constraints are defined as follows:

Hard constraint: no students are allowed to sit two or more exams simultaneously.

Soft constraint: spread the exam evenly so that students get ample time for last minute preparation before the next exams.

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

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

Where

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

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 exam k $(k \in \{1,..., N\})$

III. Graph Colouring Heuristic

Graph colouring heuristic is one of the popular sequential approaches to generate the initial solution. This method is based on ordering strategy where exam with the most 'difficulty' is chosen for scheduling first so that finally a feasible solution can be obtained. The exam difficulty is measured with various graph heuristic ordering approaches such as largest degree (LD), largest enrolment (LE), largest weighted degree (LWD), and saturation degree (SD) which have been successful in solving the examination timetabling problem [16]. In [17], authors analysed and investigated the efficiency of those four heuristics in constructing the initial solution. In [18], Roulette wheel graph colouring selection scheme was used for ordering examinations into solving the Toronto datasets. Another approach is using adaptive heuristic technique [19] where exams are ordered primarily with a particular graph heuristic and later altered according to an adaptive strategy. Similarly, other adaptive approaches were also proposed in [20][21]. Fuzzy strategy of selecting heuristic and ordering was also implemented in timetabling problem [22][23]. Recently, in [24], authors used graph heuristic to solve real world university exam timetable problem, which produced better quality solution than the university's existing software.

TABLE I. TORONTO DATASETS

Dataset	No. of Timeslots	No. of Exams	Number of Students	Conflict Density
1. car-s-91	35	682	16925	0.13
2. car-f-92	32	543	18419	0.14
3. ear-f-83	24	190	1125	0.27
4. hec-s-92	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	42	2419	30029	0.03
8. rye-s-93	23	486	11483	0.07
9. sta-f-83	13	139	611	0.14
10. tre-s-92	23	261	4360	0.18
11. uta-s-92	35	622	21267	0.13
12. ute-s-92	10	184	2750	0.08
13. yor-f-83	21	181	941	0.29

IV. Great Deluge Algorithm

Great deluge is a local search metaheuristic algorithm developed by Dueck [25]. This algorithm devises a mechanism to avoid local optima by accepting the worst solution. In the great deluge algorithm, decay rate or water level (which is a vital parameter) is used as acceptance level (or boundary). The current solution is accepted if the solution is better than the previous solution or in a certain boundary level. This boundary level controls the concentration and diversification of the search (with concentration increases in each iteration). In [26], authors investigate great deluge algorithm on Toronto datasets, and, subsequently, in [27], an extension of the earlier study was conducted by introducing flex great deluge algorithm for the examination timetabling problems. This approach was also tested on Toronto datasets and produced best results compared with other approaches cited in their literature. In a recent study, in [28], a great deluge algorithm was hybridized with other heuristic including electromagnetic-like mechanism and particle swarm optimization procedures. This approach produced competitive result when tested on Toronto. In [29], hybridization of the great deluge and artificial bee colony algorithm was proposed to solve both exam and course timetabling problem. Another recent work is observed in [4], where a great deluge algorithm was proposed for solving real world timetabling problem, namely UMP timetable datasets. The approach produced better result than authors' previous approach of graph colouring heuristics. This attracts us to further explore this algorithm in our partial exam assignment approach.

V. PROPOSED APPROACH

It is observed that most of the literatures use constructive heuristic (such as graph heuristic) to generate the initial solution follow with improvement of the initial solution using various meta-heuristic algorithms [4][12][30][31]. However, it is observed that the quality of the initial solution varies significantly from the final improved solution, with good initial solution would normally result in producing a good final improved solution [4]. In this work, we proposed partial exam assignment with improvement using great deluge algorithm. The procedure is depicted in the following subsection.

A. Graph Heuristic selection

There are many heuristic ordering sequential techniques in generating the initial solution. One of the popular techniques used in the literature is degree based ordering as they produce quality solution [32]. The graph heuristic ordering approaches widely used are as follows:

- Largest degree (LD): This technique orders the examination on the basis of largest number of conflicting examinations.
- Largest weighted degree (LWD): This heuristic order the exams based on the number of students in conflict. That is, schedule exams with most number of students who are involved in the conflict.
- Largest enrolment (LE): Here exams with the largest number of registered students are scheduled first.
- Saturation degree (SD): The exams are ordered based on the number of remaining timeslots; exams with the least number of available periods in the timetable are given higher priority to be selected for scheduling.

However, along with these heuristics we use three (3) other heuristics ordering in our experiments that are the SD (LD), SD (LWD) and SD (LE). In these ordering, the exams are first ordered according to LD, LWD or LE and the ordering will dynamically be changed according to SD ordering rules.

```
Choose heuristics and do initial ordering based on
[e.g.SD(LE),SD(LWD),SD(LE)]
Set examination assignment value
while until end of all examinations assigning in time slots do
    schedule partially exam
    calculate temporary penalty cost
    use local search algorithm for improvement
end while
if final solution vector satisfy hard constraint
  return final penalty cost as result
end if
```

Fig.1. Solving examination timetable using partial graph heuristics with improvement

B. Partial graph heuristic with improvement

The ordered exams are scheduled based on the *exam assignment value* - an integer value indicating number of exams selected for scheduling. The selected exams are scheduled to the timeslot, which are then improved using local search heuristic algorithms. In this improvement phase, modified version of great deluge algorithm has been proposed. Entire process continued until all of the exams have been scheduled. Fig. 1 shows the process of the partial exam assignment algorithm.

```
Set the initial partial solution s from the partial graph heuristic
Calculate initial cost function f(s)
n- number of neighborhood structure
Set the number of iteration I
Set initial boundary level B=f(s)
Set the desire value D
Set initial decay rate β=(B-D)/I
Set best solution s_best=s
while stopping criteria does not meet do
Calculate neighbor solutions by applying neighborhood structure
(N1,N2,N3) and consider best solution as candidate solution f(s*)
  if (f(s*)≤f(s)) or (f(s*)≤B
     s=s*
     boundary level B=B-β*random(1,5)
  end if
  if(f(s*)≤f(s_best)
     s_best=s*
  end if
  if no solution improves in M iteration
   boundary level B=B+ random (1,5)
  end if
end while
return s_best as partial best solution
```

Fig. 2. Great deluge algorithm for improvement

C. Improvement using great deluge algorithm

Great deluge algorithm has been applied to various optimization problems. In this paper, we implement a modified version of the traditional great deluge algorithm. Decay rate is initialized as β, with the *level-desire value/total number of iteration.* Three neighbourhood structures are being used (i.e. moving, swapping, transfer) and the best neighbour is taken as candidate solution for next step. Boundary level is decreased by subtracting the decay rate value. We encourage exploration by increasing the boundary level. If the solution is not improved for several iterations, such as 50 times in this case, boundary level will be increased with a random number between 1 to 5. Fig. 2 is the proposed great deluge algorithm used for improving the solution.

VI. EXPERIMENTAL SET

The 13 Toronto datasets have different number of examination; hence, the *examination assignment value* was selected based on the percentage total number of exams (i.e. 10%, 25%, 50%, and 75%). For example, 10% *exam assignment value* for 184 exams, only 18 exams are selected for scheduling. For each datasets, 30 runs using different random seeds were conducted. Neighbourhood structured used in the improvement include moving a random exam from a timeslot to a different timeslot (N_1), swapping two selected exams randomly (N_2) and swaps all exams between two timeslots selected randomly (N_3). The total number of iteration used is 100,000. However, the iteration will stop when there is no improvement after 50,000 iterations. Finally, entire program was developed in Java SE and run in core i3 PC with 2 GB RAM.

VII. RESULTS & DISCUSSION

The results of our experiment are summarised in Table II. Here, we notice that our proposed approach is able to produce quality result for all of the datasets when compared to the traditional approach of graph heuristics with great deluge algorithm as the improvement strategy. It is observed that the partial approach is able to produce and improvement for all of the datasets and able to generate as high as 17.63% improvement for car-s-91. We also found that a quality solution produce when using *small exam assignment value*. The reason is that small *exam assignment value* enables the algorithm to perform more iteration in reducing the penalty value of the objective function compared to using larger *exam assignment value*. Additionally, the SD(LD) produces the most improvement result. This is because here larger conflicted exams are given priority to be scheduled first. We believe, optimizing these conflicted exams with consideration of available timeslot for scheduling lead to reducing penalty values of the objective function.

Finally, we evaluated the proposed approach with the state-of-the-art techniques cited in literatures. Table III shows comparison of our approach with other reported result. Overall, our approach proposed in this study outperforms other results for car-s-91, car-f-92, tre-s-92, and uta-s-92 problem instances. For the rest of the datasets, our approach is able to produce competitive results compared to some of the well-known and complex algorithm. However, our approach does not differ too much with the best reported result that is below than 12% (except for rye-s-93 that is 30.5%).

The advantage of our proposed approach is that it is straight forward and requires less parameter setting. Primarily, there are two parameters needed to be tuned which is the *exam assignment value* during the graph heuristics phase and decay rate for the great deluge algorithm during the improvement phase. Furthermore, great deluge algorithm balances between the exploration and exploitation of the search space so that the quality of initial solutions can be improved.

TABLE II. COMPARATIVE RESULT BETWEEN PREVALENT GRAPH HEURISTICS WITH GREAT DELUGE AND PARTIAL GRAPH HEURISTICS WITH GREAT DELUGE

Datasets	**Traditional graph heuristic with Great deluge**		**Partial graph heuristics with Great deluge**			**Percentage of improvement %**
	Construction solution	**Great deluge**	**Great deluge**	**Graph Heuristic ordering**	**Exam Assignment value%**	
car-s-91	8.33 – LD	5.56	4.58	SD(LD)	10%	17.63
car-f-92	7.00 - LD	4.58	3.82	SD(LD)	10%	16.59
ear-f-83	52.35 - SD(LE)	33.68	33.12	SD(LD)	75%	1.66
hec-s-92	16.21 - SD(LWD)	10.53	10.32	SD(LD)	10%	1.99
kfu-s-93	23.68 - (LD)	14.79	13.34	SD(LD)	10%	9.80
lse-f-91	18.83 - (LE)	10.85	10.24	SD(LE)	10%	5.62
rye-s-93	18.28 - SD(LD)	11.39	9.79	SD(LD)	10%	14.05
sta-f-83	166.43 - SD(LE)	157.39	157.12	SD(LWD)	25%	0.17
tre-s-92	12.07- SD(LE)	8.24	7.84	SD(LD)	10%	4.85
uta-s-92	5.53 - LE	3.22	3.13	SD(LE)	10%	2.80
ute-s-92	38.03 – SD(LD)	26.96	25.28	SD(LWD)	10%	6.23
yor-f-83	49.8 - LD	35.51	35.46	SD(LWD)	50%	0.14

TABLE III. COMPARISON OF RESULT WITH STATE-OF-THE-ART APPROACHES

DataSets	**[16]**	**[21]**	**[33]**	**[34]**	**[35]**	**[36]**	**[37]**	**Our approach**	**Improvement % / difference %**
car-s-91	7.10	5.12	4.8	4.6	6.6	4.97	5.14	**4.58**	Improvement: 0.4%
car-f-92	6.20	4.41	4.1	3.9	6.0	4.28	4.70	**3.82**	Improvement: 2%
ear-f-83	36.40	36.91	34.92	32.8	**29.3**	35.86	37.86	33.23	Difference: 11.8%
hec-s-92	10.80	11.31	10.73	10.0	**9.2**	11.85	11.90	10.32	Difference:10.8%
kfu-s-93	14.00	14.75	**13.0**	13.0	13.8	14.62	15.30	13.34	Difference:2.5%
lse-f-91	10.5	11.41	10.01	10.0	**9.6**	11.14	12.33	10.24	Difference:6.25%
rye-s-93	7.3	9.61	9.65	–	**6.8**	9.65	10.71	9.79	Difference: 30.5%
sta-f-83	161.5	157.52	158.26	**156.9**	158.2	158.33	160.12	157.12	Difference: 0.1%
tre-s-92	9.6	8.76	7.88	7.9	9.4	8.48	8.32	**7.84**	Improvement:0.5%
uta-s-92	3.5	3.54	3.2	3.2	3.5	3.40	3.88	**3.13**	Improvement: 2.23%
ute-s-92	25.8	26.25	26.11	24.8	**24.4**	28.88	32.67	25.28	Difference: 3.5%
yor-f-83	41.7	39.67	36.22	**34.9**	36.2	40.74	40.53	35.46	Difference:1.6%

*The best results is highlighted in **bold**

VIII. Conclusions and Future Work

In this paper, we have presented partial graph heuristic with great deluge algorithm to solve the examination timetabling problem. The partial approach uses the *exam assignment value* for selecting exams for scheduling and improvement (with great deluge algorithm). We evaluate our algorithm using Toronto benchmark datasets. From the analysis, it is observed that the partial graph heuristic with great deluge algorithm produces better results than the traditional approach. Finally, the proposed method is generally able to produce competitive results when compare to other reported result from the literature and also outperforms other results for some of the datasets. For future works, we will implement the algorithm on the ITC2007. We are also optimistic that the solution can be further improved by applying other meta-heuristic approaches (e.g. simulated annealing , tabu search and so on).

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] 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.

[7] N. Mansour, V. Isahakian, and I. Ghalayini, "Scatter search technique for exam timetabling," *Applied Intelligence,* vol. 34, pp. 299-310, 2011/04/01 2011.

[8] 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.

[9] 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.

[10] S. Abdullah, H. Turabieh, B. McCollum, and P. McMullan, "A Tabu-Based Memetic Approach for Examination Timetabling Problems," in *Rough Set and Knowledge Technology*. vol. 6401, J. Yu, S. Greco, P. Lingras, G. Wang, and A. Skowron, Eds., ed: Springer Berlin Heidelberg, 2010, pp. 574-581.

[11] N. R. 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.

[12] K. Anwar, A. T. Khader, M. A. Al-Betar, and M. A. Awadallah, "Harmony Search-based Hyper-heuristic for examination timetabling," in *Signal Processing and its Applications (CSPA), 2013 IEEE 9th International Colloquium on*, 2013, pp. 176-181.

[13] E. Burke, G. Kendall, M. Mısır, and E. Özcan, "Monte Carlo hyper-heuristics for examination timetabling," *Annals of Operations Research,* vol. 196, pp. 73-90, 2012/07/01 2012.

[14] 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.

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

[16] 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.

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

[18] N. R. Sabar, M. Ayob, G. Kendall, and R. Qu, "Roulette wheel graph colouring for solving examination timetabling problems," in *Combinatorial Optimization and Applications*, ed: Springer, 2009, pp. 463-470.

[19] 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.

[20] A. Soghier and R. Qu, "Adaptive selection of heuristics for assigning time slots and rooms in exam timetables," *Applied Intelligence,* vol. 39, pp. 438-450, 2013/09/01 2013.

[21] 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.

[22] 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.

[23] H. Asmuni, E. K. Burke, J. M. Garibaldi, and B. McCollum, "Fuzzy multiple heuristic orderings for examination timetabling," in *Practice and Theory of Automated Timetabling V*, ed: Springer, 2005, pp. 334-353.

[24] 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.

[25] 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.

[26] E. Burke, Y. Bykov, J. Newall, and S. Petrovic, "A time-predefined local search approach to exam timetabling problems," *IIE Transactions,* vol. 36, pp. 509-528, 2004.

[27] E. K. Burke and Y. Bykov, "Solving exam timetabling problems with the flex-deluge algorithm," 2006.

[28] H. Turabieh and S. Abdullah, "An integrated hybrid approach to the examination timetabling problem," *Omega,* vol. 39, pp. 598-607, 12// 2011.

[29] C. W. Fong, H. b. Asmuni, B. McCollum, P. McMullan, and S. Omatu, "A new hybrid imperialist swarm-based optimization algorithm for university timetabling problems," *Information Sciences,* 2014.

[30] N. R. Sabar, M. Ayob, G. Kendall, and R. Qu, "A honey-bee mating optimization algorithm for educational timetabling problems," *European Journal of Operational Research,* vol. 216, pp. 533-543, 2/1/ 2012.

[31] M. Ayob12, E. K. Burke, and G. Kendall, "An iterative re-start variable neighbourhood search for the examination timetabling problem," *Practice and Theory of Automated Timetabling VI,* p. 336, 2006.

[32] E. K. Burke, B. McCollum, A. Meisels, S. Petrovic, and R. Qu, "A graph-based hyper-heuristic for educational timetabling problems," *European Journal of Operational Research,* vol. 176, pp. 177-192, 2007.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.