2013 IEEE 9th International Colloquium on Signal Processing and its Applications, 8 - 10 Mac. 2013, Kuala Lumpur, Malaysia

Harmony Search-based Hyper-heuristic for Examination Timetabling

Khairul Anwar[1], Ahamad Tajudin Khader[1], Mohammed Azmi Al-Betar[1,2], Mohammed A. Awadallah[1]
[1] School of Computer Sciences, Universiti Sains Malaysia (USM), 11800
Pulau Pinang, Malaysia
[2] Department of Computer Science, Jadara University, Irbid-Jordan
Email: ka10_com097@student.usm.my

***Abstract*—In this paper we proposed a Harmony Search-based Hyper-heuristic (HSHH) method for examination timetabling problems. The Harmony Search Algorithm (HSA) is a relatively new metaheuristic algorithm inspired by the musical improvisation process. The Hyper-heuristic is a new trend in optimization that uses a high level heuristic selected from a set of low-level heuristic methods. Examination timetabling is a combinatorial optimization problem which belongs to NP-hard class in almost all of its variations. In HSHH approach, the HSA will operate at a high level of abstraction which intelligently evolves a sequence of improvement low-level heuristics to use for examination timetabling problem. Each low-level heuristics represents a move and swap strategies. We test the proposed method using ITC-2007 benchmark datasets that has 12 *de facto* datasets of different complexity and size. The proposed method produced competitively comparable results.**

Keywords: Examination Timetabling Problems; Harmony Search Algorithm; Hyper-heuristic methods

I. INTRODUCTION

Timetabling problems arise in many fields which are related to services especially in hospitals or clinics (such as nurse or surgeon timetabling), transportation (such as bus, train or airplane timetabling) and also in educational institutions (such as school, college or university timetabling). Timetabling can be defined as a process of assigning certain resources or events to the limited timeslots (and rooms) according to a set of constraints.

The timetabling problem can be classified as an NP-complete problem [1] which could become extremely difficult when it involves a large number of events or resources (can be a hundred or thousand) to be scheduled and wide variety of constraints which need to be satisfied or taken into consideration [2]. This kind of problem can be solved through the correct method or approach which makes this entire field become more effective and efficient. Based on a survey [2], the most studied area for the timetabling problem is educational timetabling which includes school timetabling and university timetabling (course and exam). This research will focus on the university examination timetabling problem.

Examination timetabling is the process of scheduling given exams to given timeslots and rooms in accordance with given *hard* and *soft* constraints. The hard constraints must be satisfied while the soft constraints are desired but not absolutely necessary. The main target is to find an examination timetabling solution that satisfies all hard constraints and minimizes the violation soft constraints as much as possible.

The interests of the researchers have been drawn to the problem (examination timetabling problems) for many years and several techniques and approaches have been produced and applied, such as sequential techniques, constraints based techniques, local search based techniques, population based techniques, and hyper-heuristics (see in [2]).

Recently, there has been a growing trend toward more general methods where one of the approaches is Hyper-heuristic (HH). Hyper-heuristics is referred as *"using heuristic to choose heuristic"* and this methodology actually tries to find the best heuristic to solve the optimization problem. The main difference between the hyper-heuristics and meta-heuristics is that hyper-heuristics operate on the search space of heuristics (or heuristic component) rather than directly on the search space of solution. The main purpose of applying the hyper-heuristics is to build general method or algorithm that can solve different versions of timetabling problems. This approach intends to increase the level of generality in which optimization methodologies can operate. There are several researchers applying hyper-heuristics in solving examination timetabling problem, such as Tabu search [3],[4], variable neighborhood search [5],[6], graph based methods [7],[8], monte carlo [9], scatter search [10] and genetic programming [11].

Harmony search algorithm (HSA) is a population-based algorithm developed by Geem, [12]. HSA is a stochastic search mechanism, simple in concept, and no derivation information is required in the initial search [12]. It has been successfully applied to a wide range of optimization and scheduling problems such as course timetabling [13], nurse scheduling [14] and examination timetabling problem [15]. Though HSA has been successful in solving the examination timetabling problem, we are working on a different approach in solving the same problem. The main objective of this research is to investigate and introduce the harmony search algorithm within hyper-heuristic framework in solving the examination timetabling problems which is known as

Harmony Search Hyper-Heuristic (HSHH). For the evaluation purpose, one of the competition tracks in the Second International Timetabling Competition 2007 (ITC-2007), which is the examination timetabling track, is used as a benchmark dataset. Results from the evaluation showed that the proposed HSHH is comparable to the previous works.

This paper is organized as follows: Section II discusses the examination timetabling problem (ETP) and benchmark dataset of ITC-2007 for examination track. The details of harmony search hyper-heuristic (HSHH) algorithm is presented in Section III. Section IV discusses the computational results and comparison with other results that have been published. Finally, the conclusion and future works are presented in Section V.

II. EXAMINATION TIMETABLING PROBLEM

This research focuses on the examination timetabling problem. Basically in examination timetabling, the problem of assigning a set of exams into a limited number of time periods and rooms which are free from clashes or conflict [16]. During assigning the exams, it must satisfy a set of constraints which is known as *hard* and *soft* constraints. Hard constraints must be satisfied at all times. The solutions that satisfied all the hard constraints are called *feasible solutions*. Soft constraints are more desirable as these constraints can be violated if necessary, but should be avoided if possible. Soft constraints are normally used to measure the quality of the solutions. The examination timetabling problem could be more complicated or difficult when the constraints that need to be satisfied increases and sometimes contradict with each other [2].

In this research, ITC-2007 datasets is used as a benchmark [17]. ITC-2007 datasets is a capacitated examination timetabling benchmark datasets which has eight problem instances. In these datasets, there are several hard and soft constraints that need to be satisfied. The following shows the hard and soft constraints as provided by ITC-2007 dataset.

Hard Constraints

- H1: No student sits more than one examination at the same time;
- H2: The capacity of any examination room is not exceeded at any time throughout the examination session;
- H3: Period Lengths are not violated;
- H4: Satisfaction of period related hard constraints (e.g. exam B must be scheduled after exam A);
- H5: Satisfaction of room related hard constraints (e.g. exam A exclusively scheduled in room X);

Soft Constraints

- S1: Two exams in a row;
- S2: Two exams in a day;
- S3: Specified spread of examinations;
- S4: Mixed duration of examinations within individual periods;
- S5: Larger examinations appearing later in the timetable;
- S6: Period related soft constraints – some period has an associated penalty;
- S7: Room related soft constraints – some room has an associated penalty;

The main objective of solving this dataset problem is to satisfy all the hard constraint and minimize the violation or penalty value of the soft constraint as given in expression (1) in order to produce a quality timetable [17].

$$min\ f(\boldsymbol{x}) = \sum_{s \in S} (w^{2R} C_s^{2R} + w^{2D} C_s^{2D} + w^{PS} C_s^{PS}) + w^{NMD} C^{NMD} + w^{FL} C^{FL} + C^P + C^R \quad (1)$$

Where $\boldsymbol{x}$ is a complete timetabling solution; S is referring to a set of students, w refers to the institutional weights for each constraint except for period and room related soft constraint ($C^P and\ C^R$). Table 1 shows the detailed notation of (1).

TABLE 1: List of penalties related to ITC-2007 soft constraints.

Math Symbol	Description
C_s^{2R}	"Two exams in a row" penalty for student *s*
C_s^{2D}	"Two exams in a day" penalty for student *s*
C_s^{PS}	"Period spread" penalty for student *s*
C^{NMD}	"No mixed duration" penalty
C^{FL}	"Front load" penalty
C^P	"Period" penalty
C^R	"Room" penalty

III. HARMONY SEARCH HYPER-HEURISTICS

In this research, the harmony search is implemented at a higher level of abstraction rather than being applied directly to the solution space which is similar to the previous work carried out by [18], [19], [20], [3] and [10]. In this approach, two types of neighborhood structures have been employed as low-level heuristics. They are summarized as follows:

- $\boldsymbol{h1}$: select two exams at random and swap the timeslots.
- $\boldsymbol{h2}$: select one exam at random and move to a new randomly selected feasible timeslot.

Algorithm 1 shows the pseudo-code of the Harmony Search Hyper-heuristics (HSHH) that have developed which has five main steps as follow:

Step 1: Initialization. The HSHH begins with setting the harmony search parameter: harmony memory size (HMS) similar to population size in GA, harmony memory consideration rate (HMCR) and number of iterations (NI). Here we also introduce a new parameter to control Heuristic Harmony Memory (HHM) called Harmony Memory Length (HML). Then the initial feasible solutions *(S_F)* will be generated using constructive heuristics (i.e., Largest Degree) [16]. These heuristics have been chosen because it can produce a feasible initial solution [4],[8],[13]. If the solutions are not feasibly completed, the repairing methods will be used. In this work, two different repairing methods are proposed. First repairing method is to assign the unscheduled exam into timeslots that are feasible to them which the selected timeslot have the lowest conflict with other exams. In second repairing method, all exams will be rescheduled but in this time, the unscheduled exams will be scheduled first. This process will be repeated until the feasible solution is produced or the stop criterion is met.

Step 2: Initializing of Harmony Memory. In hyper-heuristic environment, HSHH consists of two different search spaces or harmony memory which are Heuristic Harmony Memory (HHM) and Solution Harmony Memory (SHM). This work is different from previous work conducted in [15]. Here HSA is used on the heuristics search space instead of solutions search space, where in Genetic Algorithm (GA), it is known as *indirect chromosome* [18],[10]. HHM contains sets of heuristic vectors determined by HMS where every vector is a heuristics sequence (i.e. *h'*) and the length of the sequences is determined by HML. Example of HHM is as shown in Figure 1.

Heuristic sequence *h'*

h'	1	1	2	2	1	1

Heuristic Harmony Memory (HHM)

h^0	1	1	2	1	2	1
h^1	1	2	1	1	1	2
...						
...						
h^{HMS}	1	2	2	2	1	2

Note that 1 represent low-level heuristic *h*1 and 2 represent low-level heuristic *h*2

Fig 1: Example of HHM

In initializing HHM and SHM, the HSHH first construct the heuristics sequence (*h'*) randomly and apply this random sequence to the initial feasible solution (S_F). The new solution (*s'*) will be evaluated using the objective function (*f(s')*) as in (1). This process will be repeated until HHM and SHM is filled (see (2) and (3)).

Step 3: Improvise a new heuristic HM. In this step, a new heuristics sequence ($h' = h'_1, h'_2, \ldots . . h'_{HML}$) is generated from scratch based on two operators: memory consideration and random consideration. In this work, we are not concerned about the pitch adjustment operator in the improvisation step as we only employed two types of neighborhood structures as low-level heuristics.

$$HHM = \begin{bmatrix} h_1^1 & h_2^1 & \cdots & h_{HML}^1 \\ h_1^2 & h_2^2 & \cdots & h_{HML}^2 \\ \vdots & \vdots & \ddots & \vdots \\ h_1^{HMS} & h_2^{HMS} & \cdots & h_{HML}^{HMS} \end{bmatrix} \quad (2)$$

$$SHM = \begin{bmatrix} s_1^1 & s_2^1 & \cdots & s_N^1 \\ s_1^2 & s_2^2 & \cdots & s_N^2 \\ \vdots & \vdots & \ddots & \vdots \\ s_1^{HMS} & s_2^{HMS} & \cdots & s_N^{HMS} \end{bmatrix} \quad (3)$$

Note that *N* refers to the number of examinations.

In *memory consideration operator*, the new value of h'_1 is randomly selected from the historical values (e.g.$h_1^1, h_1^2, \ldots . . h_1^{HMS}$), and stored in the heuristic harmony memory with probability of HMCR where $0 \le HMCR \le 1$ as shown in (4). For *Random consideration operator*, the new value is randomly assigned according to the available heuristic range (*H* = {*h*1, *h*2}) with probability of (1-HMCR) as in (4).

$$h'_i = \begin{cases} \in \{h_1^1, h_1^2, \ldots . . h_1^{HMS}\} & w.p. \quad 0 \le HMCR \le 1 \\ \in \{h1, h2\} & w.p. \quad (1 - HMCR) \end{cases} \quad (4)$$

The new harmony of heuristic sequence *h'* will be applied to a solution (S^{Rand}) that is randomly selected from SHM. Here the HSHH randomly select the solution from the SHM to avoid the local optima.

Step 4: Update HHM and SHM. The new solution (*s'*) will be evaluated using the objective function. The new solution must be complete and feasible. In some iteration, the heuristic sequence may not improvise a complete and feasible solution. If this happen, the heuristics sequence will be rejected and new sequence will be generated. If the new solution is better than the worst solution in harmony memory, the new heuristics sequence *h'* and new solution *s'* will be saved in the memory (*h'* in HHM and *s'* in SHM) and the worst heuristic sequence and solution will be excluded from the memory (i.e., HHM and SHM).

Step 5: Check the stop criterion. Step 3 and step 4 in this approach are repeated until the stop criterion (NI) is met. NI represents the number of iterations.

Algorithm 1: The Pseudo-code of Harmony Search Hyper-Heuristics (HSHH).

Step 1: Initialization

- Set HSHH parameters (HMCR, NI, HMS and new parameter HML).
- Find initial feasible solution (S_F).

Step 2: Initialize Harmony Memory

- Initialize the HHM and SHM.
- Recognize the worst ($f(s^{worst})$) and the best ($f(s^{best})$) vector for HHM and SHM.

Step 3: Improvise a new Heuristic Harmony Memory

- *h'*= 0; //vector of heuristic sequences
- for l = 0,…,HML do
- if (U(0,1) < HMCR) then
- Memory consideration
- else
- Random consideration
- end if
- end for
- Select random solution from SHM (S^{Rand}).
- Apply the heuristic sequence *h'* to the selected solution S^{Rand}.

Step 4: Update HHM and SHM

- If ($f(s') < f(s^{worst})$) then
- save heuristics sequence *h'* and new solution *s'* the memory (HHM and SHM)
- exclude worst heuristic sequence and solution from the memory
- end if

Step 5: Check the stop criterion

Repeat step 3 and 4 (Improvise and update) until the stop criterion is met (NI).

IV. Experiments And Results

In this section, Harmony Search Hyper-heuristic is evaluated using the standard benchmark dataset (ITC-2007) for the university examination timetabling problem. Table 3 shows details of the dataset. The proposed method is coded in Microsoft Visual C++ 6 under Windows 7 on Intel processor with 2GB RAM. We chose to test the proposed method with each problem instance in ITC-2007. We ran the experiment 10 times for each problem instance due to the stochastic nature of the method. The Harmony Search Hyper-Heuristic (HSHH) parameters setting is as shown in Table 2. These values were experimentally decided.

TABLE 2: HSHH Parameters setting

Parameter	Values
Heuristic HMS	10
Solution HMS	10
HMCR	0.99
Heuristic HML	10
NI	100000

TABLE 3: The Characteristics of ITC-2007 Examination Timetabling Dataset

Dataset	Info1	Info2	Info3	Info4	Info5
Dataset1	7891	7833	607	54	7
Dataset2	12743	12484	870	40	49
Dataset3	16439	16365	934	36	48
Dataset4	5045	4421	273	21	1
Dataset5	9253	8719	1018	42	3
Dataset6	7909	7909	242	16	8
Dataset7	14676	13795	1096	80	15
Dataset8	7718	7718	598	80	8
Info1: Number of students; *Info2*: Actual students; *Info3*: Number of Exams; *Info4*: Number of timeslots; *Info5*: Number of rooms;					

The best results produced by HSHH are shown in Table 4,which also featured the results from the ITC-2007 winner [8],[21] for comparison.

TABLE 4: Comparison results for HSHH with ITC-2007 winner

Dataset	HSHH	Muller (2009)	Gogos et. al. (2008)	Atsuta et. al. (2007)	De Smet (2008)	Pillay (2007)
Dataset1	11823	**4370**	5905	8006	6670	12035
Dataset2	976	**400**	1008	3470	623	3074
Dataset3	26770	**10049**	13862	18622	-	15917
Dataset4	-	**18141**	18674	22559	-	23582
Dataset5	6772	**2988**	4139	4714	3847	6860
Dataset6	30980	**26950**	27640	29155	27815	32250
Dataset7	11762	**4213**	6683	10473	5420	17666
Dataset8	16286	**7861**	10521	14317	-	16184

Table 5 presents the comparison between HSHH with other available result that have been published.

TABLE 5: Comparison with other methods and approaches that have been published

Dataset	HSHH	M1	M2	M3	M4	M5	M6
Dataset1	11823	4775	4370	4633	6234	6582	**4368**
Dataset2	976	**385**	**385**	405	395	1517	390
Dataset3	26770	**8996**	9378	9064	13002	11912	9830
Dataset4	-	16204	**15368**	15663	17940	19657	17251
Dataset5	6772	**2929**	2988	3042	3900	17659	3022
Dataset6	30980	**25740**	26365	25880	27000	26905	25995
Dataset7	11762	4087	4138	**4037**	6214	6840	4067
Dataset8	16286	7777	7516	**7461**	8552	11464	7519

The results compared in Table 5 are described as follows:

- **M1**:[22] – An improved multi-staged algorithmic
- **M2**:[23] – A Three phase constraint-based approach
- **M3**:[24] – An extended great deluge algorithm
- **M4**:[8] – A graph coloring constructive Hyper-heuristic
- **M5**:[21] – Artificial Bee Colony algorithm
- **M6**:[25] – Hybrid approach incorporates effective heuristics operators within great deluge algorithm

The results produced by HSHH may not be the best result, but they are comparable to the results produced by previous work as shown in Figure 2 and interestingly at certain point in the iterations, the heuristics sequence can make very big improvements to the solutions as shown in the Figure 3. This would be the focus for future research that is to enhance the performance of the algorithm and to further improve the results.

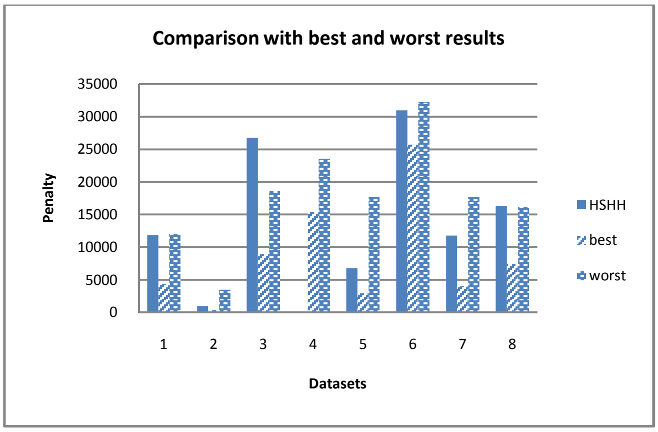

Fig 2: Comparison of HSHH with best and the worst results that have been published

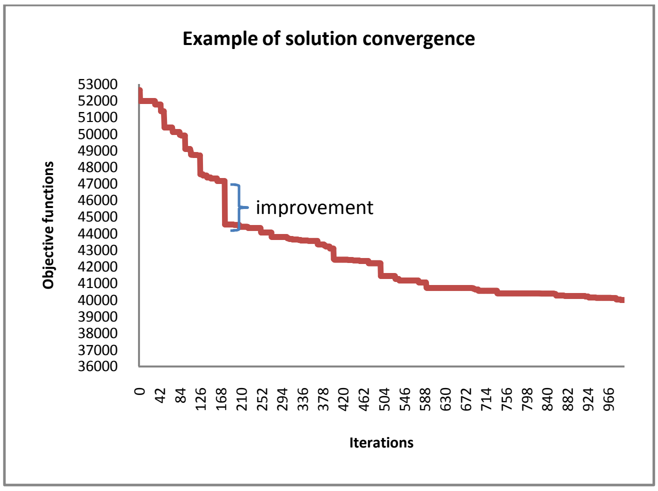

Fig 3: The improvement of solution at certain point of the iterations

V. Conclusion And Future Directions

This paper presented Harmony search-based hyper-heuristics (HSHH) for solving examination timetabling problems using ITC-2007 datasets. This algorithm employed HSA at the high level to evolve a sequence of improvement low-level heuristics. At the low level, two different neighborhood structures are used. By testing HSHH on each of the problem instances of ITC-2007 dataset, the experimental result showed that HSHH are able to solve examination timetabling problem. As this is an initial investigation of harmony search in hyper-heuristic framework, the results produced by HSHH in this study are not desirably comparable with the best results that have been published.

For future directions: to increase the number of low-level heuristics instead of only two low-level heuristics and to include the pitch adjustment operator in the improvisation step. We also plan to adopt learning mechanism within the HSHH algorithm in order to improve the heuristic selection and to enhance the speed of convergence.

References

[1] R. Lewis, "A survey of metaheuristic-based techniques for university timetabling problems," *OR Spectrum,* vol. 30, pp. 167-190, 2008.

[2] R. Qu, E. 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.

[3] E. K. Burke, G. Kendall, and E. Soubeiga, "A tabu-search hyperheuristic for timetabling and rostering," *Journal of Heuristics,* vol. 9, pp. 451-470, 2003.

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

[5] R. Qu and E. Burke, "Hybrid variable neighborhood hyperheuristics for exam timetabling problems," in *Meta-heuristic International Conference*, Vienna, 2005.

[6] 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, 2010.

[7] 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.

[8] N. R. Sabar, M. Ayob, R. Qu, and G. Kendall, "A graph coloring constructive hyper-heuristic for examination timetabling problems," *Applied Intelligence,* pp. 1-11, 2011.

[9] E. K. Burke, G. Kendall, M. Mısır, and E. Özcan, "Monte carlo hyper-heuristics for examination timetabling," *Annals of Operations Research,* pp. 1-18, 2012.

[10] N. R. Sabar and M. Ayob, "Examination timetabling using scatter search hyper-heuristic," in *2nd Conference on Data Mining and Optimization, 2009 (DMO '09)*, Universiti Kebangsaan Malaysia, Bangi, Selangor, 2009, pp. 127-131.

[11] N. Pillay and W. Banzhaf, "A genetic programming approach to the generation of hyper-heuristics for the uncapacitated examination timetabling problem," in Neves J, Santos MF, Machado J (Ed.), *Progress in Artificial Intelligence, 13th Portuguese Conference on Artificial Intelligence*, 2007, pp. 223-234.

[12] Z. W. Geem, J. H. Kim, and G. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation,* vol. 76, pp. 60-68, 2001.

[13] M. A. Al-Betar and A. T. Khader, "A harmony search algorithm for university course timetabling," *Annals of Operations Research,* pp. 1-29, 2012.

[14] M. A. Awadallah, A. T. Khader, M. A. Al-Betar, and A. L. Bolaji, "Nurse Scheduling Using Harmony Search," in *Sixth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA)*, 2011, pp. 58-63.

[15] M. A. Al-Betar, A. T. Khader, and F. Nadi, "Selection mechanisms in memory consideration for examination timetabling with harmony search," in *Proceedings of the 12th annual conference on Genetic and evolutionary computation (GECCO '10)*, New York, USA, 2010, pp. 1203-1210.

[16] M. Carter and G. Laporte, "Recent developments in practical examination timetabling," *Practice and Theory of Automated Timetabling,* pp. 1-21, 1996.

[17] B. McCollum, P. McMullan, E. K. Burke, A. J. Parkes, and R. Qu, "The second international timetabling competition: Examination timetabling track," Technical Report QUB/IEEE/Tech/ITC2007/Exam/v4. 0/17, Queens University, Belfast (UK)2007.

[18] H. Terashima-Marín and P. Ross, "Evolution of constraint satisfaction strategies in examination timetabling," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'99)*, 1999.

[19] P. Ross, S. Schulenburg, J. G. Marín-Blázquez, and E. Hart, "Hyper-heuristics: learning to combine simple heuristics in bin-packing problems," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2002, pp. 942-948.

[20] L. Han and G. Kendall, "An investigation of a Tabu assisted hyper-heuristic genetic algorithm," in *Congress on Evolutionary Computation (CEC '03)*, Canberra, Australia, 2003, pp. 2230-2237 Vol. 3.

[21] M. Alzaqebah and S. Abdullah, "Artificial bee colony search algorithm for examination timetabling problems," *International Journal of the Physical Sciences,* vol. 6(17), pp. 4264-4272, 2 September, 2011.

[22] C. Gogos, P. Alefragis, and E. Housos, "An improved multi-staged algorithmic process for the solution of the examination timetabling problem," *Annals of Operations Research,* vol. 194, pp. 203-221, 2012.

[23] T. Müller, "ITC2007 solver description: a hybrid approach," *Annals of Operations Research,* vol. 172, pp. 429-446, 2009.

[24] B. McCollum, P. McMullan, A. Parkes, E. Burke, and S. Abdullah, "An extended great deluge approach to the examination timetabling problem," *Proceedings of the 4th Multidisciplinary International Scheduling: Theory and Applications 2009 (MISTA 2009),* pp. 424-434, 2009.

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