2013 IEEE 3rd International Conference on System Engineering and Technology, 19 - 20 Aug. 2013, Shah Alam, Malaysia

Harmony Great Deluge for Solving Curriculum Based Course Timetabling Problem

Juliana Wahid
Universiti Utara Malaysia
Kedah, Malaysia
w.juliana@uum.edu.my

Naimah Mohd Hussin
Universiti Teknologi MARA
Perlis, Malaysia
naimahmh@perlis.uitm.edu.my

***Abstract*—University course timetabling which has been determined as non deterministic polynomial problem that accept widely as problem that are intractable. An efficient algorithm does not exist that is guaranteed to find an optimal solution for such problems. The design of good algorithm to find new methods and techniques to solve such problem is a very active area of research. This paper presents the adaption of the hybridizing between harmony search with great deluge algorithm for solving curriculum-based course timetabling problems. The algorithm can be adapted to the problem. Results were not comparatively better than those previously known as best solution. Proper modification in terms of the approach in this algorithm would make the algorithm perform better on curriculum-based course timetabling.**

***Index Terms*—Harmony Search, Great Deluge, Curriculum Based Course Timetabling**

I. Introduction

University course timetabling which has been determined as non deterministic polynomial (NP) problem that accept widely as problem that are intractable, i.e there does not exist an efficient algorithm that is guaranteed to find an optimal solution for such problems [1]. This is parallel with the theorem of *No Free Lunch Theorem* [2] that states if any prior assumptions cannot be made about the optimization problem we are trying to solve, no algorithm can be expected to perform better than any other algorithm on that problem. The design of good algorithm to find new methods and techniques to solve such problems is a very active area of research.

A very promising research issue is the hybridization of meta-heuristics techniques [3] and there are different forms of hybridization from this issue. The first one consists of including components from one meta-heuristic into another one. The second form concerns systems that are sometimes labeled as cooperative search. They consist of various algorithms exchanging information in some way. The third form is the integration of approximate and systematic (or complete) methods.

The works done in this paper, fall in the first category in which the great deluge (GD) will be included as one operator in the harmony search (HS) in solving the curriculum-based course timetabling. This is merely a work that hybridizes the population-based method, i.e. harmony search with the local search method, i.e. great deluge. HS deals in each iteration with a set of solutions rather than with a single solution. This lead to better in identifying promising areas in the search space, whereas GD are better in exploring promising areas in the search space [3].

Previous works that similar with this category of hybridization are such from Al-Betar & Khader [4], Abdullah et. al [5] and Bolaji et. al [6]. In their work, Betar & Khader hybridizing the harmony search with hill climbing optimiser (HCO) and particle swarm optimization (PSO) focussed in solving the post-enrolment course timetabling problems. Abdullah et. al. focused on investigating a genetic algorithm and sequential local search while Bolaji et. al applied the population based artificial bee colony (ABC) algorithm. Both works in [5] and [6] focused in solving the curriculum-based course timetabling problems.

This paper is organized as follows; section two provides the detailed description of CB-CTT in terms of hard and soft constraints; details of Harmony Great Deluge (HGD) are presented in section three; the ways in which HGD was adapted for CB-CTT are discussed in section four. Section five shows the computational results and their comparison with previous findings in the literature. In section five also, several consideration for future enhancement are discussed. The final section presents the conclusion.

II. Curriculum-Based Course Timetabling Problem

The university course timetabling is the tasks that assign the set of courses offered by the university to particular rooms and time slots. The assigning process needs to satisfy several constraints so that timetable can actually be carried out i.e. feasible. There are two categories of constraints in timetabling such as hard and soft constraints [7]. Hard constraints are constraints that cannot be violated, e.g. courses must be allocated into different time slots in order to avoid students and staffs clashes. While soft constraints are constraints that reflect the preferences and teaching/learning comfort expectations of the students and staffs which are not necessarily vital. Examples of soft constraints are avoiding students having to attend three or more courses in successive time slots, and preventing students from having only one course in any day.

978-1-4799-1030-4/13/$31.00 ©2013 IEEE

The university course timetabling as described in International Timetabling Competition 2007 (ITC-2007) [www.cs.qub.ac.uk/itc2007/] fall into two versions. The first version is the post enrolment course timetabling (PE-CTT), in which the timetable is constructed based on student enrolments after students have selected which lectures to attend. The second version is the curriculum-based course timetabling (CB-CTT). This version will construct the timetable according to curricula published by the university. This paper focusing on the CB-CTT and the details of CB-CTT are discussed as follows:

A. Problem Description

The CB-CTT has:

- A set of N courses, $e = \{e_1,\ldots,e_N\}$, each course is composed of L same lectures to be scheduled and associated to a teacher.
- A set of P periods, $T=\{t_1,\ldots t_p\}$.
- A set of M rooms, $R=\{R_1,\ldots R_M\}$.

The CB-CTT problems deal with the assignment of a set of lectures of courses to a set of rooms and timeslots on a weekly basis, in accordance with a given set of constraints. A period p is a pair composed of a day and a timeslot. Thus, the total number of scheduling periods is the product of the number of days and the number of timeslots per day. In addition, there is a set of q curricula $CU = \{cu_1, cu_2, \ldots, cu_q\}$ where each curriculum cu_i corresponds to a group of courses such that any pair of courses in the group have students in common.

A timetable is considered feasible once all lectures of courses have been assigned to timeslots and rooms with respect to the hard constraints. In addition, a feasible timetable satisfying the four hard constraints gives a penalty cost for the violations of the four soft constraints. The main objective of the CB-CTT problem is to minimize the number of soft constraint violations in a feasible solution.

The following are the hard constraints:

H1- Lectures: All lectures of a course must be assigned to a distinct period and a room.

H2 - Room Occupancy: Two lectures cannot be scheduled to the same room and the same period.

H3 - Conflicts: Lectures of courses in the same curriculum or taught by the same teacher must be scheduled to different periods.

H4 - Availability: If the teacher of a course is not available at a given period, then no lectures of the course can be allocated to that period.

The soft constraints are outlined below:

S1 - Room Capacity: The number of students attending the course for each lecture must be less than or equal to the capacity of the rooms hosting the lectures.

S2 - Minimum Working Days: The lectures of each course should be spread across a given number of days.

S3 - Curriculum Compactness: Lectures of courses belonging to the same curriculum should be in consecutive periods (i.e., adjacent to each other).

S4 - Room Stability: All lectures of a particular course should be assigned to the same room; otherwise, the number of occupied rooms should be less.

The quality of solution is calculated as the total penalties of the soft constraints: S1 + S2 + S3 + S4.

III. The Hybrid of Harmony Search and Great Deluge

A. Harmony Search (HS)

The harmony search (HS) algorithm is a new metaheuristic algorithm that impersonates the musical improvisation process in which a group of musicians improvise their instruments' pitch by searching for a perfect state of harmony according to audio-aesthetic standard [8]. Geem [9] identified that HS contains parameters such as:

- Harmony memory size (HMS) - number of solution vectors concurrently handled.
- Harmony memory considering rate (HMCR) – ($0 \leq$ HMCR ≤ 1) where HSA picks one value randomly from memory.
- Pitch adjusting rate (PAR) - ($0 \leq$ RAR ≤ 1) is the rate where HSA fine-tunes the value which was originally picked from memory.
- Maximum improvisation (MI) - number of iterations

Al-Betar, et al. [10] stated that HMS is similar to population size, HMCR similar to crossover rate in genetic algorithm, PAR used to determine the number of courses which will be moved to another position or swapping them with other courses, MI is similar to number of iterations in optimisation algorithms.

The HS consists of several steps such as 1) problem formulation, 2) algorithm parameter setting, 3) random tuning for memory initialization, 4) harmony improvisation (random consideration, memory consideration, and pitch adjustment), 5) memory update, 6) performing termination.

B. Great Deluge (GD)

The great deluge (GD) algorithm proposed by [11] requires two items of data to be specified in advance, i.e. water level and decay rate. This meta-heuristic always accepts better solutions than the one that it is currently considering. It will accept *worse* solutions if the evaluation function value is less than (or equal) to a particular water level. This is lowered during the run (according to a *decay rate*). The decay rate is a simple function of the two parameters. It is defined as the evaluation value of the initial solution subtracted by the estimate of the desired evaluation value, all divided by the desired number of moves (computational expense). The flow chart of the GD can be seen in Fig 1.

In GD, the water level is set to a value higher than the expected penalty of the best solution at the start of the search. Then the water level is decreased in a linear fashion during the search until it reaches a value of zero. There are many variants form of GD in the literature such as non linear great deluge (NLGD), extended great deluge (EGD) and flex deluge (FD).

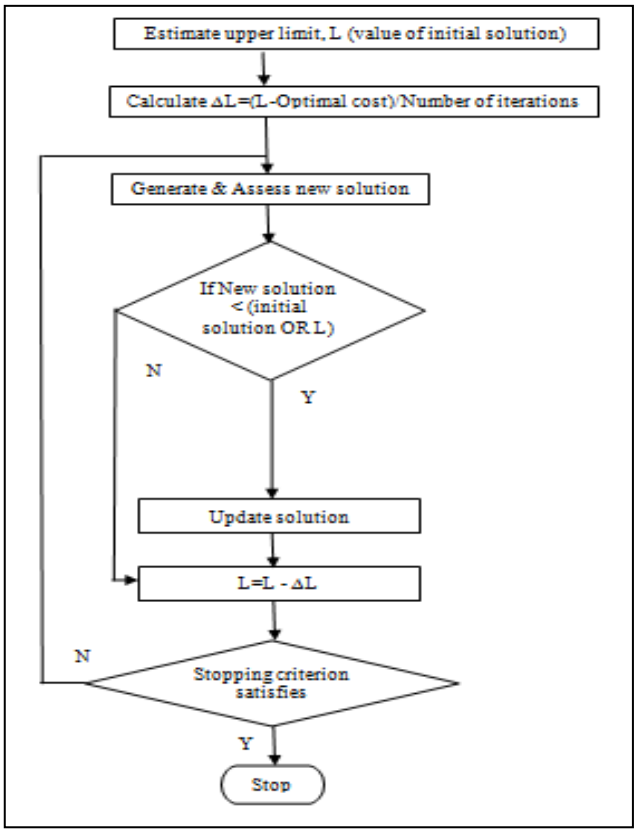

Fig. 1. Flow Chart of Great deluge

This paper was the first attempt to hybridize between HS with GD at the time the study was conducted. The original GD will be used in this work.

C. Hybridization of HS and GD

HS can identify promising areas in a set of solutions whereas GD are better in exploring promising areas in the single solution. Harmony great deluge (HGD) maintained the HS core mechanism, while the GD is used to improve the random consideration outside HM as shown in Fig 2. In HGD the initial solution is acquired randomly from HM that contains population of initial solutions. The selected initial solution will be used in GD procedure that shown in Fig 1. At the end of GD procedure, if the new solution produced is better (lower cost) than the highest cost in the HM, it then will be inserted to the HM and solution with the highest cost in the HM will be deleted. Otherwise, the iteration starts all over again executing the HS steps.

IV. HGD FOR CB-CTT

The approach to adapt the HGD to CB-CTT is divided into two stages: constructive phase and improvement phase.

A. Constructive Phase

The construction phase generates a feasible initial solution satisfying all the hard constraints. This approach started from an empty timetable until feasible timetable achieved which fulfil all the hard constraints. The population of number of initial solutions and corresponding objective function need to be generated because the harmony search is a population-based algorithm. The constructive approach uses graph heuristic in which courses were ordered by saturation degree followed by largest degree. In [12], this setting produced the maximum number of elements in population of initial solution i.e. 50 which equal to number of iterations except for problem instances comp05 (only 13 initial solutions) and comp12 (only 49 initial solutions).

To ensure that problem instances to have the same number in the population of initial solution, i.e. 50, the iteration in producing the initial solution were increased for problem instances comp05 and comp12. Table 1 shows the detail for the number of iterations together with the total initial solutions produced. The best 50 initial solutions from the total will be used in the improvement phase.

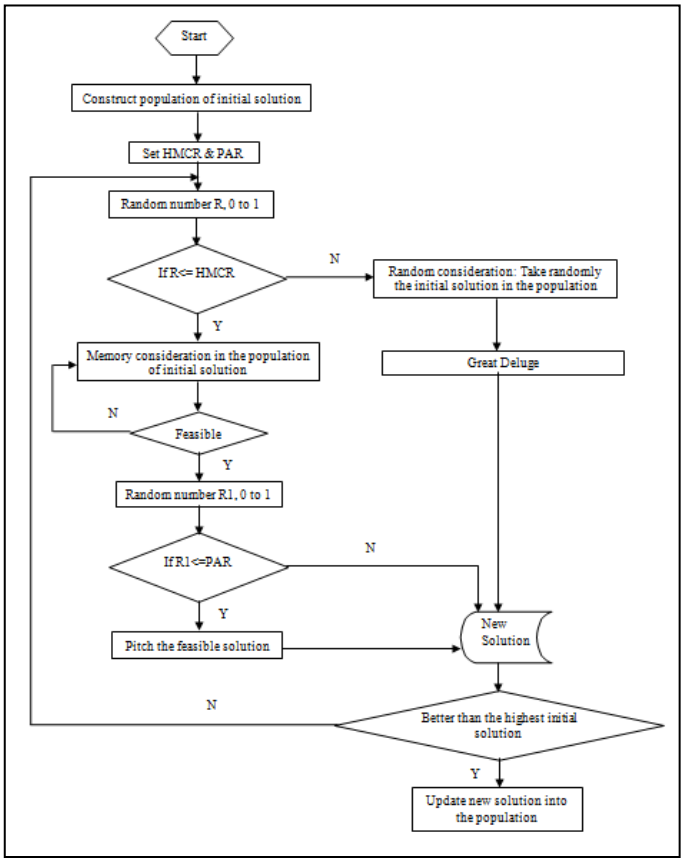

Fig. 2. Flow Chart of HGD

TABLE I. SEVERAL POPULATION OF INITIAL SOLUTION FOR COMP5 AND COMP12

PROBLEM INSTANCES	NO OF ITERATION	TOTAL	MIN	MAX
COMP05	50	13	1297	2173
	200	34	1296	2321
	300	46	1296	2373
	500	**70**	**1242**	**2373**
COMP12	50	49	1542	1919
	60	**58**	**1542**	**1954**

B. Improvement Phase

In this stage, the HGD was adapted in searching for an optimal solution with respect to the soft constraint penalties by generating new solution.

HS consists of several steps as stated in section III. Specifically, the improvement phase relates directly to step number 4, i.e. harmony improvisation. In this particular step, there are three operators to be implemented, i.e. memory consideration operator, pitch adjustment operator, and random consideration operator.

In memory consideration, the value of the course number for new solution was selected from all other course number located in the same column of the HM. This research used the occurrence of the course number in the HM column with least available period that randomly scheduled in a new harmony with probability of HMCR. If the operators cannot find a feasible timetable (this happens in some medium and large data instances), the algorithm initiates a repair process to find feasible timetable, such as find an empty timeslot for unassigned course number and swapping between course numbers.

Once the feasible timetable produced, with probability of PAR, the timetable went through the pitch adjustment operator. Two pitch adjusting procedures were designed, each one of which reflects a neighbourhood structure i.e. move and swap the courses between timeslot.

Memory consideration was carried out based on the probability of HMCR, the random consideration was applied if the probability is 1-HMCR. The GD procedures took place in which the initial solution taken randomly from the HM. The neighbourhood search used were move and swap between course numbers.

V. EXPERIMENTAL RESULTS

In this section, the performance of HGD algorithm for solving the CB-CTT problem was evaluated using 14 data instances generated in ITC-2007. The details of the problem can be found in http://tabu.diegm.uniud.it/ctt/index.php. The proposed method is coded using C++ and executed in Microsoft Visual 2008 under Windows Vista on an Intel Machine with Core TM and a 2.16GHz processor and 1GB RAM.

Table II shows the experimental results of HGD algorithm. The first column is the name of the data instances, the second column represents the initial solution at construction phase, the third column shows the results of HGD while the last two columns shows the best known solution.

TABLE II. EXPERIMENTAL RESULTS OF HGD

Problem instances	Initial Solution	HGD	Best Known Solution[*] (Until 5/10/2012)	
	Penalty	*Penalty*	*Penalty*	*Method Used*
Comp01	323	251	5	Tabu Search
Comp02	747	692	24	SAT-based
Comp03	715	671	66	Local Search
Comp04	692	659	35	Local Search
Comp05	1297	1181	290	SA
Comp06	982	926	27	SAT-based
Comp07	1063	1021	6	SAT-based
Comp08	788	765	37	Other
Comp09	849	811	96	Tabu Search
Comp10	920	910	4	SAT-based
Comp11	215	184	0	Tabu Search
Comp12	1542	1368	300	SA
Comp13	818	756	59	Tabu Search
Comp14	720	720	51	Mathematical Programming

*http://tabu.diegm.uniud.it/ctt/index.php

Number of penalty in column 3 shows that HGD can improve the CB-CTT from its initial solution (column 2) for most of the data instances except for Comp14. The approach used in this study is still not able to achieve results comparable with the best known solution. An improvement (which is currently ongoing) is required, in order to enhance the performance of the proposed method to produce desired outcomes better than those currently in use.

VI. DISCUSSIONS

There several issues that needed to be considered in enhancing the approach such as the memory consideration operator, the pitch adjustment operator and the random consideration operator.

In the memory consideration operator, instead of courses that ordered by least available period, perhaps other selection schemes can be incorporated. Al-Betar et. al [13] carried out experiment that involved several selection mechanisms used to selects variable (courses) from a solution in the HM in memory consideration operator. The other selection mechanisms consists of Global best Harmony Search (GHS), Proportional Harmony Search (PHS), Tournament Harmony Search (THS), Linear rank Harmony Search (LHS), and Exponential rank Harmony Search (EHS).

In pitch adjustment operator and random consideration operator, specifically the GD method, there are other kinds of neighborhood search that can be considered such as the Kempe chain neighbourhood and S-chains that can be explored and applied in the algorithm.

VII. CONCLUSIONS

This paper presented the harmony great deluge for solving the CB-CTT problems. Results from the experiments have shown that the algorithm is capable of solving timetabling problems. Although the results produced by the algorithm in this study are presently not comparatively better than those already reported in the literature, this is initial adaptation of HGD. Further works need to be considered to improve the HGD performance.

REFERENCES

[1] C. P. Gomes and R. Williams, "APPROXIMATION ALGORITHMS," in *SEARCH METHODOLOGIES Introductory Tutorials in Optimization and Decision Support Techniques*, G. Kendall and E. K. Burke, Eds., ed: Springer Science-i-Business Media, LLC, 2005, pp. 557-585.

[2] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization " *Evolutionary Computation, IEEE Transactions on Volume: 1 , Issue: 1 Digital Object Identifier: 10.1109/4235.585893 Publication Year: 1997 , Page(s): 67 - 82* 1997.

[3] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization:Overview and conceptual comparison," *ACM Computing Surveys, 35(3):268-308, 2003,* 2003.

[4] M. Al-Betar and A. T. Khader, "A hybrid harmony search for university course timetabling," in *Multidisciplinary International Conference on Scheduling : Theory and Applications (MISTA 2009) 10-12 August 2009, Dublin, Ireland*, 2009.

[5] S. Abdullah*, et al.*, "An Investigation of a Genetic Algorithm and Sequential Local Search Approach for Curriculum-based Course Timetabling Problems," *Multidisciplinary International Conference on Scheduling : Theory and Applications (MISTA 2009) 10-12 August 2009, Dublin, Ireland,* 2009.

[6] A. L. Bolaji*, et al.*, *Tackling University Course Timetabling Problem Using Artificial Bee Colony Algorithm. CHAPTER TWELVE . ISBN 978-969-9742-00-2.*, 2012.

[7] E. K. Burke*, et al.*, "Automated university timetabling: The state of the art. ," *The Computer Journal, 40(9), 565–571.,* 1997.

[8] Z. W. Geem*, et al.* (2001) A new heuristic optimization algorithm: harmony search. . *Simulation 76, 60–68 (2001)*. Available: http://sim.sagepub.com/content/76/2/60.abstract

[9] Z. W. Geem, "State-of-the-Art in the Structure of Harmony Search Algorithm," in *Recent Advances In Harmony Search Algorithm*. vol. 270, Z. Geem, Ed., ed: Springer Berlin / Heidelberg, 2010, pp. 1-10.

[10] M. Al-Betar*, et al.*, "A harmony search algorithm for university course timetabling," in *PATAT 2008*, 2008.

[11] G. Dueck, "New Optimization Heuristic: The Great Deluge Algorithm and the Recordto-record Travel.," *Journal of Computational Physics 104, 86–92,* 1993.

[12] W. Juliana and M. H. Naimah, "Constructing Population of Initial Solutions for Curriculum-Based Course Timetabling Problem," in *International Conference on Innovation, Management and Technology Research (ICIMTR2012), 21-22 May 2012*, Melaka, Malaysia, 2012.

[13] M. A. Al-Betar*, et al.*, "Novel selection schemes for harmony search," *Applied Mathematics and Computation,* vol. 218, pp. 6095-6117, 2012.