A Comparative Study of EGD Based Hyper Heuristic for ETP

Ei Shwe Sin[1], Nang Saing Moon Kham[2]
Research Student[1], Head of Department of Information Science [2]
University of Computer Studies, Yangon
eishwe.ucsy@gmail.com,

Abstract

This paper makes a comparative study of the Extended Great Deluge (EGD) based hype heuristic (HH) for the university exam timetabling problem. Indeed, Hyper-Heuristic is not new in AI field. It is an emerged search technology to select or generate (new) low level heuristics for combinatorial optimization problems. In the general framework of HH, it has two main stages: heuristic selection and move acceptance. For the latter stage, most of Meta heuristic algorithms are used. EGD has been firstly proposed and also used in HH as our previous job. Based on the numerous well-known papers and our previous experience, now, we investigate again the proposed EGD based HH to make the analysis of its performance and comparable with other methods in the literature or not. As another contribution for more comparison, Simple Random, Variants of Great Deluge(GD):Non linear GD(NLGD) and Flex Deluge(FD) are also employed in EGD based HH by applying it on the ETP.

1. Introduction

In recent years, hyper-heuristics have been increasingly used to solve a wide range of optimization problems. Though research work on hyper-heuristics was started from the 1960's, the term hyper-heuristic has only been introduced recently [1], [3]. A hyper-heuristic is a process which, when given a particular problem instance and a number of low-level heuristics, manages the selection of which low level heuristic to apply at any given time, until a stopping condition is met [1], [4], [16]. Burke et al, (2009a) made a well clear classification of HH approaches. With respect to the nature of the heuristic search space, two popular groups can be identified. They are constructive and perturbative (local search) hyper-heuristics. This paper is included into the latter and University of Exam Timetabling Problem (ETP) is used as case study.

The rest of the paper is organized as follows: the Section 2 reviews hyper heuristic approaches for ETP while Section 3 describes the detail description of ETP .The Section 4 discusses the proposed EGD based in detail. Experimental analysis and results have revealed in Section 5. Finally, the conclusion is shown at the Section 6.

2. Related Work

Although there are many techniques to exam timetabling problem, in this section, we only review about hyper heuristic approach from the literature. Recent years have witnessed the great success of hyper-heuristics applying to exam timetabling problem. Nelishia Pillay [17] reported on the use of an evolutionary algorithm (EA) to search a space of heuristic combinations for the uncapacitated examination timetabling problem.

In [18], he also investigated using a genetic programming based hyper-heuristic system to evolve heuristic combinations for the uncapacitated examination timetabling problem. The performance of the genetic programming based system using the different representations is applied to three examination timetabling problems with different characteristics and the performance on these problems is compared. The results obtained are also compared to that of other hyper-heuristic systems applied to the same problems.

In 2007, Bilgin et al. also reported that a simple random-great deluge based hyper heuristic was the second best after choice function-simulated annealing, considering the average performance of all hyper heuristic over a set of examination timetabling problems [1]. In [10, 11] and [16], K.Graham and M. H. Naimah investigated a Tabu search based hyper-heuristic and applied it for solving examination timetabling problems. In 2010, Ender Ozcan et. al has proposed the great deluge based hyper heuristic with reinforcement learning(RL) for exam timetabling problem [13]. Moreover, in 2009, the EGD algorithm is also investigated and made a comparison with the first winner, Tomas Muller in the 2nd International Timetabling Competition (ITC2007). And it seems that EGD is comparable to existing state of the art techniques, and form previous application to other data sets and a different problem domain (course timetabling)[2].

Therefore, by taking the advantages of EGD and referring [13, 2] , in [14] ,we proposed it to firstly used in HH to avoid the continuous lack of improvement of the problem solution, by cooperating with RL and applied it to the Toronto Benchmark Datasets. It is also compared and reported the results.

3. Case Study: Exam Timetabling Problem Description and Datasets

In general exam timetabling problem, it consists of assigning a set of exams into a limited number of timeslots while satisfying a set of constraints: Hard Constraints that cannot be violated in any circumstances; and Soft constraints which should be satisfied as much as is possible. In this paper, the following hard constraints are used:

- No exams with common resources (e.g. students) assigned simultaneously.
- Resources of exams need to be sufficient (i.e. size of exams need to be below the room capacity, enough rooms for all of the exams.)
- Each examination must be assigned to a timeslot only for once.
- All the examinations must be scheduled.

If the timetable solution is not violating the defined hard constraints, it can be called a feasible solution. As the soft constraint,

- A student should have at least a single timeslot in between his/her examinations in the same day.

The basic task of this problem has similarities across very different institutions although the requirements and needs can differ markedly [6]. The variants of this problem are capacitated and uncapacitated. To represent the exam timetabling problem, there are also three types of models. They are Graph Model, Mathematical model, Object oriented model (UML) model. This paper used the mathematical model to describe the problem as follow:

E: The number of **n** exams: $E_1, E_2, E_3,\dots, E_n$

S: The number of **m** students: $S_1, S_2, S_3,..,S_m$

T: The number of **k** timeslots: T1, T2,T3,….Tk

B_{n*k} A binary matrix such that bik=1 when exam ei is assigned to the timeslot t € T and bik =0 otherwise.

C= $(c_{ij})_{n*n}$ The conflict matrix; where each element (denoted by cij where i, j) is the number of students that have to take both exams i and j. This is a symmetrical matrix of size N, where diagonal element c_{ij} equal the number of students who have taken exam i.

The objective is to schedule all of the exams into time slots, while minimizing the average total cost per student. The following function is used to calculate the average cost per student.

$$\frac{\sum_{i=1}^{n} w_i (| e_i - e_j |) c_{ij}}{S} \quad (1)$$

In equation, w_i is the weight that represents the importance of scheduling exams with common students i timeslots apart , where, w(1)=16, w(2)=8 w(3)=4,w(4)=2 and w(5)=1, i.e. the smaller the distance between periods the higher the weight allocated. Note for n>5, w (n) =0. For example, if a student has two consecutive examinations (i.e. no free time between them) then the weight value of 16 is assigned. If a student has two consecutive exams with a free timeslot in between then a value 8 is assigned and so on. The value of c_{ij} is the number of students common to both examinations.

To the best of our knowledge, for ETP, most of the researchers have used the Toronto benchmark dataset, which can be available at http://www.asap.cs.nott.ac.uk/resources/data.sht ml. The given table 1 presents the characters of the dataset. In each problem instance, the two files will be included. They are

(1) **Course files** - It is ended with the extension "**.crs** "or "**.exm**". It shows the number of exam and enrolment of student for it.

(2) **Student files**-It is ended with extension "**.stu**" .The second one displays the students list who takes which exams.

Table 1. Characteristics of Toronto Benchmark Dataset

Instance	Exam	Enrolment	Density	Period
Car91 I	682	56877	0.13	35
Car92 I	543	55522	0.14	32
Ear83 I	190	8109	0.27	24
Hecs92 I	81	10632	0.42	18
Kfu93	461	25118	0.06	20
Ise 91	381	10918	0.06	18
Pur93 I	2419	120681	0.03	42
Rye 92	486	45051	0.07	23
Sta83I	139	5751	0.14	13
Tre92	261	14901	0.18	23
Uta92	622	58979	0.13	35
Ute92	184	11793	0.08	10
Yor83I	181	6034	0.29	21

There is no information related with invigilators/lecturers in the data instances of Toronto dataset. In addition, although room capacity is showed, how many rooms are there and how many students/seats are allowed in each room are lacked in each data instance.

4. Proposed EGD based HH

This section discusses the proposed EGD based HH. The given figure 1 shows the system flow chart of the proposed EGD based HH. In figure, there are five steps and these steps will be iterated according to the maximum iteration, which is set 1000. The related works in each step will be discussed in the following sections respectively.

4.1 Loading Problem and Creating Initial Solution

To create an initial solution, it is not necessary to be completely feasible. However, it is preferred to be as feasible as possible because the quality of initial solution would affect the final solution.

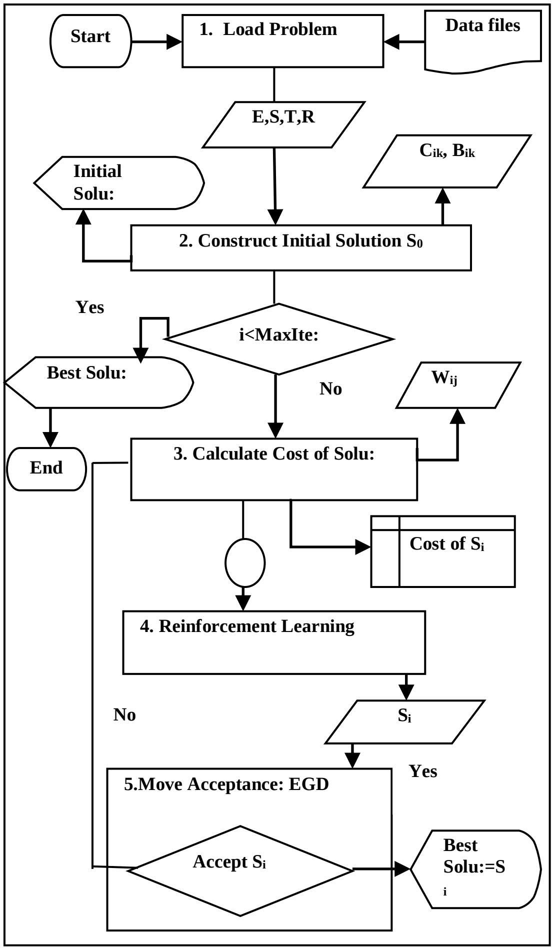

Figure 1. System Flow Chart of the Proposed EGD based HH

Therefore, in this paper, completely feasible solution is produced by using (LE) heuristic. The examination with the largest student enrolment is selected and kept in the vector. And then another exams which are not conflict with the exam in the vector, are added to that vector. This process is repeated until the required capacity for exams in the vector is less than the total capacity and unscheduled exam list is empty. Finally, this exam vector which satisfied all hard constraints is assigned to the timeslot sequentially. Here, we used the average number of exams per timeslot to be balanced in assignment of exam-timeslots.

After constructing the initial solution or candidate solutions, the costs of them are needed to calculate according to the objective function.

4.2 Reinforcement Learning

Machine learning techniques are vital to make the right choices for the heuristic selection process in hyper heuristics. Most of the existing online learning hyper heuristic incorporates reinforcement learning (RL). It is a sub-field of machine learning, represents an important direction for research in Artificial Intelligence [15]. RL is a framework for learning an optimal policy of a task from trials. It requires less a priori knowledge.

The following low level heuristics will be chose by RL:

- Swap Timeslot with Kempe Chain (ST-KC): swapping a subset of exams in two distinct timeslots making sure that a hard constraint violation does not occur.
- Reassign Timeslot (RT): randomly reassign a sequence of timeslots.
- Inverting Timeslot (IT) - inverting timeslots making sure that a hard constraint violation does not occur.
- Shifting Timeslot (ST) -left or right shifting a sequence of timeslots.

For each of them, utility value is assigned value one as initial. The values are allowed to change within an interval of [0, number of heuristics] and lower bound and upper bound of them are set 0 and 40 respectively. In this paper, not only a simple **P:1-N:1** (Additive adaption-Negative adaption) strategy to increase and decrease the utility value but also maximum utility method are used for RL[15].

4.3 Move Acceptance: Extended Great Deluge (EGD) and Solution Output

Due to the advantages of GD such as only need to tune a few parameters, it has been widely used. However, it can cause the continuous lack of improvement which means the final solution is same with the initial after the complete execution, which can lead to RL to select only one heuristic repeatedly. To overcome it, the original GD is extended by adding reheat mechanism rather than termination with improvement. In this paper, we use the half-life decay rate, is the amount of time it takes for half of the amount of substance to decay to attempt to reach the optimal solution. The wait parameter is used to invoke the reheat mechanism. It can be specified in terms of percentage or number of total moves in the process [2]. For the representation of the initial and final solution, there are two forms of assignment. They are

- Exam-Timeslot assignment
- Exam- Classroom assignment.

In this paper, the first one is used as the follow:

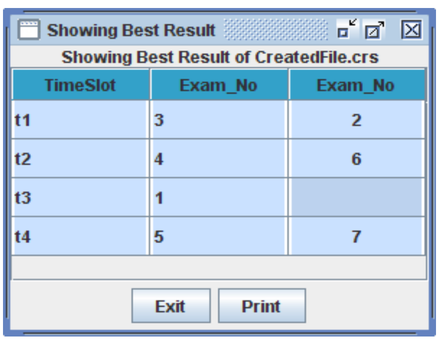

Figure 2. Example of Output of the optimal Solution

It can be seen that the results of HH by using EGD in the next section for the case study.

5. Results and Discussion

Our experimental analysis is making on the computer Pentium IV, Dell with the RAM 2GB. The proposed HH is evaluated over an arbitrarily selected subset of the Toronto benchmark datasets and other 18 data instances created by random generator. It is also compared to the EGD based HH combining with Simpler Random (SR) as shown in Table 1 and 2.

Each data instance is executed 10 times and each run is performed starting from the same initial configuration. The Lowest Best Cost is used as the performance criterion for all experiments. According to the objective function, the lowest the cost, the better the timetable is. Furthermore, the percentage improvement introduced by an approach over another one is computed by using the average quality of solutions over ten runs for a given problem instance. Standard Deviation is also used as another criterion.

Table 2. Standard Deviation of proposed HH and SR-EGD

Datasets	**RL-EGD**	**SR-EGD**
Car-f-92	**0.1017**	0.1029
Hec-s-92	**0.3586**	0.38873
Rye-s-93	**0.0404**	0.0789
Lse-f-91	0.2423	**0.0994**
Sta-f-83	**0.9490**	0.9613
Tre-s-92	**0.4529**	0.0627
Uta-s-92	0.0477	**0.0426**
Ute-s-92	**0.3313**	0.5026
Yor-f-83	**0.3346**	0.3943

Table 3. Comparison of RL-EGD and SR-EGD

Datasets	Comparison of two HHs	% Impro
Car-f-92	RL-EGD ≥ SR-EGD	0.12
Hec-s-92	RL-EGD ≥ SR-EGD	0.65
Rye-s-93	RL-EGD ≥ SR-EGD	3.2
Lse-f-91	RL-EGD ≥ SR-EGD	1.4
Sta-f-83	RL-EGD ≥ SR-EGD	0.56
Tre-s-92	RL-EGD ≥ SR-EGD	20.9
Uta-s-92	RL-EGD ≥ SR-EGD	6
Ute-s-92	RL-EGD ≥ SR-EGD	2.22
Yor-f-83	RL-EGD ≥ SR-EGD	0.15

The given tables 2 and 3 show the comparison results of RL-EGD and SR-EGD. In table 1, the proposed RL-EGD has smaller standard deviation value (i.e. Bold entries) than SR-EGD in most of the data instances; expect Lse-f-91 and Uta-s-92. Table 2 presents the percentage improvement uses the average best cost obtained in ten runs for SR-EGD as the baseline.

Moreover, to make a more comparison, we also implemented other two HHs as the alternative ways, by using variants of GD: Non-linear GD and Flex Deluge employing as move acceptance method in HH. The results of lowest best cost for each HH methods are presented in the following graphs.

The figure 3 shows the comparison for 5 data instances from the Carter's Datasets. We choose these data instances by randomly. In the figure, X axis shows the data instances names while Y axis is being representing the lowest cost of the best solution. Here, it can produce much lower costs in three instances such as **Tre92, Ise91 and Ear83I** than those by other two HH methods. It can be observed that RL-EGD can provide the lowest best cost although there is a slight difference in the cost values of three HH methods in each dataset.

The Student's T-Test is also performed additionally to assure the induction of the comparison. Table 4 also reports comparison between proposed method's results and best results obtained for each data set in the literature, by citing 22 papers published between 1996 and 2010. In that table, the second column shows the highest result whereas the third one presents the lowest best result from the literature. It has achieved the lowest best cost in all of the data sets except Hecs92I, in which the cost value is 14.1 and it is higher than [16].

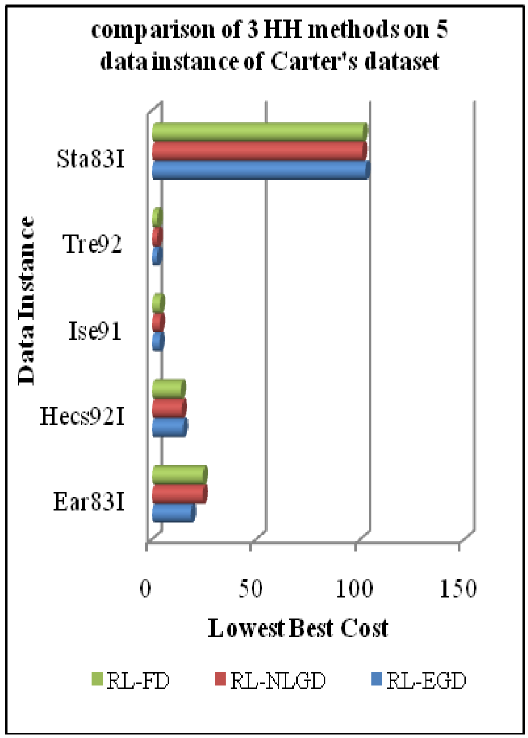

Figure 3. Lowest best Cost Comparison of 3 HHs for arbitrary data instances from Toronto Dataset

However, it is obvious that the proposed EGD based HH can comparable to the other methods in the literature.

Table 4. Comparison of results obtained from proposed EGD based HH and the literature

Instance	Highest Cost	Lowest Cost	Proposed Approach
Car91 I	6.9 [22]	3.81 [7]	2.68
Car92 I	7.1 [8]	4 [21]	3.1
Ear83 I	45.8 [16]	29.3 [9]	23.49
Hecs92 I	**12.9** [16]	9.2 [9]	**14.1**
Kfu93	18 [10]	12.81 [7]	3.239
Ise 91	15.5 [10]	9.6 [9]	4
Pur93 I	9.65 [18]	3.9 [8]	3.87
Rye 92	14 [22]	6.8 [9]	6.14
Sta83I	168.73[4]	134.9 [22]	116.37
Tre92	10 [10]	7.72 [7]	2.5
Uta92	4.5 [22]	2.88 [3]	2.78
Ute92	32 [22]	22.94 [9]	15.84
Yor83I	44.1 [22]	34.78 [8]	30.86

6. Conclusion

In this paper, a proposed EGD based HH for solving ETP has been presented. From analysis of Table 2, the proposed RL-EGD is better than SR-EGD. According to the smaller value of Standard Deviation in Table 3, it is obvious that it is stable approach for ETP. And it can also produce much lower costs in three instances than those by other two HH methods. However, when the student's T-Test is performed, it is found that there is no statistically significant between these three HHs: RL-EGD, RL-NLGD and RL-FD. The reason is that it is needed to increase the sample size. Therefore, we plan to redesign for the T-test and make analysis again as future work. However, because of the reheat mechanism of the RL-EGD, it can also produce the lowest costs in most of the data instances.

References

[1] B. Bilgin, E. Ozcan, & Korkmaz E. E.(2007), "*An experimental study on hyper heuristic and exam timetabling*", In(PATAT'06)(LNCS 67,pp.394-412).

[2] B. McCollum, P.J. McMullan, A. J. Parkes, E.K. Burke, S. Abdullah, "*An Extended Great Deluge Approach to the Examination Timetabling Problem* " , MISTA,2009.

[3] Burke, E.K., McCollum, B., Meisels, A., Petrovic, S., Qu, R.: "*A Graph-Based Hyper-Heuristic for Educational Timetabling Problems",* European Journal of Operational Research 176, 177–192 (2007).

[4] Burke, E.K., Newall, J.: "*Enhancing timetable solutions with local search methods"*. In: Burke, E.K., De Causmaecker, P. (eds.) PATAT 2002. LNCS, vol. 2740, pp. 195–206. Springer, Heidelberg (2003).

[5] Burke, E.K., Kingston, J., de Werra, D.: In: Gross, J., Yellen, J. (eds.) "*Applications to timetabling. Handbook of Graph Theory"*, pp. 445–474. Chapman Hall/CRC Press (2004).

[6] Burke, E.K., Eckersley, A.J., McCollum, B., Petrovic S., Qu, R.: *Hybrid variable neighbourhood approaches to university exam timetabling*. Accepted by European Journal of Operational Research, Technical Report NOTTCS-TR-2006-2, School of Computer Science, University of Nottingham, United Kingdom (2009).

[7] Burke, E. K, Yuri Bykov, " *A Late Acceptance Strategy in Hill Climbing for Exam Timetabling Problems*"

[8] Carter, M.W., Laporte, G., Lee, S.Y.: *Examination timetabling: Algorithmic strategies and applications*. Journal of Operational Research Society 47(3), 373–383 (1996).

[9] Caramia, M., Dell'Olmo, P., Italiano, G. F.: *New algorithms for examination timetabling.* In: Naher, S., Wagner, D. (eds.) WAE 2000. LNCS, vol. 1982, pp. 230-241. Springer, Heidelberg (2001).

[10] Di Gaspero, L., Schaerf, A.: *Tabu search techniques for examination timetabling*. In: Burke, E.K., Erben, W. (eds.) PATAT 2000. LNCS, vol. 2079, pp. 104–117. Springer, Heidelberg (2001).

[11] E. Burke, G. Kendall, E. Soubeiga. "*A tabu-search hyperheuristic for timetabling and*

roistering", Journal of Heuristic, 9:451-470, 2003.

[12] E. Soubeiga, "*Development and application of hyper-heuristics to personnel scheduling*," PhD. Dissertation, School of Computer Science, University of Nottingham, UK, 2003.

[13] E. Ozcan, M. Misir, G. Ochoa, E. K. Burke , "*A reinforcement learning- great deluge hyper heuristic for Examination Timetabling*" , In: Journal of Applied Met heuristic Computing, 1(1), 39-59, January-March 2010.

[14] ES Sin, "*Reinforcement Learning with EGD based Hyper Heuristic System for Exam Timetabling Problem*", In: Proceedings of the IEEE-CCIS 2011.

[15] Gabriela Serban , " *A new reinforcement learning algorithm*", STUDIA UNIV. BABES-BOLYAI, INFOMATICA, Volume XLVIII, Nubmer 1,2003.

[16] Kendall, G. and Hussin, N.M.: *Tabu Search Hyper-Heuristic Approach to the Examination Timetabling Problem at University Technology MARA*. In: Burke, E.K., Trick, M. (eds.) PATAT 2004. LNCS, vol. 3616, pp. 199-218. Springer, Heidelberg (2005).

[17] N Pillay, "*Evolving hyper heuristics for the uncapacitated examination timetabling problem*", Journal of Operational Research Society 63,47-58

[18] Nelishia Pillay and Wolfgang Banzhaf, " *A genetic programming approach to the generation of hyper heuristics for the uncapacitated examination timetabling problem*", EPIA 2007, LNAI 4874, pp. 223-234, 2007 @ Springer-Verlag Berlin Heidelberg 2007

[19] Merlot, L.T.G., Borland, N., Hughes, B.D., Stuckey, P.J.: *A hybrid algorithm for the examination timetabling problem*. In: Burke, E.K., De Causmaecker, P. (eds.) PATAT 2002. LNCS, vol. 2740, pp. 207–231. Springer, Heidelberg (2003).

[20] Nasser R. Sabar • Masri Ayob • Graham Kendall, "*Solving Examination Timetabling Problems using Honey-bee Mating Optimization (ETP-HBMO)*", Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA 2009)10-12 August 2009, Dublin, Ireland.

[21] Nasser R. Sabar • Masri Ayob • Graham Kendall, "*Tabu Exponential Monte Carlo with Counter Heuristic for Exam Timetabling*", 978-1-4244-2757-4/09/$25.00 ©2009 IEEE

[22] R. Abounacer, J.Boukachour, B. Dkhissi, A. El Hilali Alaoui, "*A hybird ACO with the Complete Local Search with Memory approach*", Revue ARIMA, vol.12(2010), pp. 15-42.