EGD Based Hyper Heuristic System for Scheduling Problem

Ei Shwe Sin, Nang Saing Moon Kham.
University of Computer Studies, Yangon
eishwe.ucsy @gmail.com, moonkhamucsy@gmail.com

Abstract

Nowadays, scheduling problems such as employee scheduling, university timetabling, arise in almost all areas of human activity. In the literature, there are many methods to solve it. Some of the most effective techniques on the benchmark data in the literature are Meta heuristic. However, these methods depend upon parameter tuning or the way of embedding domain knowledge .As a result, they are not capable of dealing with other different problems .Therefore, this has led to the development of hyper heuristics system. In this paper, we propose the extended great deluge (EGD) method as a move acceptance method to drive the selection of low level heuristic within hyper heuristic framework. It is applied to a benchmark set of examination timetabling problem as an instance of a constraint based real world optimization problem.

1. Introduction

Hyper heuristic is a high level problem solving methodology that performs a search over the space generated by a set of low level heuristics. It has been seen an increase number of successful implementations for problems such as producing scheduling, timetabling, personnel scheduling and other related ones. According to the literature review, there are several hyper heuristic approaches have been proposed to solve the exam timetabling problem.

Exam timetabling is one of the most important administrative activities that take place several times a year in all academic institutions. The basic task of that has similarities across very different institutions although the requirements and needs can differ markedly [8]. As the difficulty of the problem increases and their importance in practice and inherent scientific challenge, they have been widely investigated across both the operational research and the artificial intelligence community.

Early research works on hyper heuristic focused on the development of advanced strategies for choosing the heuristics to be applied at different points of the search [1].Likewise, researchers have proposed different acceptance criteria to drive the selection of low level heuristics within a hyper heuristic framework. For instance, Ayob and Kendall used a Monte Carlo acceptance criterion while Kendall and Mohamad used the great deluge acceptance criterion [2].

In this paper, an extended great deluge (EGD) based hyper heuristic system is proposed by applying it to the Toronto benchmark exam timetabling problem. The rest of the paper is organized as follows: the section 2 reviews the previous methods that are related our proposed system while Section 3 describes the exam time tabling problem. In Section 4, the proposed extended great deluge hyper heuristic method is presented. Finally, conclusions and future research is the subject of Section 5

2. Related Work

A vast amount of research has been conducted in the domain of examination timetabling with various methodologies being applied to the problem in an attempt to produce better quality timetables. The significantly larger list of publications as well as more detailed information about examination timetabling studies can be found in [10, 8, 11, and 12]. Qu et al. [6] provided the most recent survey in 2009.

For the comprehensive specific review of hyper heuristic approach; see a recent survey in [8, 9, and 13]. Actually, these methods have been present in the literature since the early 1960s. It was motivated by the awareness of many different approaches to different problems, each with different relative performances, particular advantages and individual flaws.

Typically, a hyper heuristic can conduct with a single point or multi-point search. A single iteration of a hyper heuristic method can be decomposed in two stages, heuristic selection and movement acceptance. The second one is emphasized in this paper. In general, the movement acceptance can be deterministic or nondeterministic. There are many methods which are used as move acceptance criteria in hyper heuristic. Among them, the great deluge, non linear great deluge algorithms and simulated annealing methods are very popular.

In 2003, Bykov Y. proposed the time-predefined great deluge algorithm and Trajectory base search to exam timetabling [10]. In 2006, Edmund K. Burke and Yuri Bykov made an extension of the great deluge algorithm (which they called "Flex-Deluge") where the acceptance of uphill moves depends on a "flexibility" coefficient, for solving exam timetabling problem. Good results were presented and they suggested that the flex deluge method is relatively higher effective in the large-scale problems [5].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 [14]. In recent, for course timetabling problem, non linear great deluge algorithm (NLGD) was proposed by Landa-Silva and Obit. That method produced new best in 4 out of 11 course timetabling problem instances of datasets [3]. Recently, Ender Ozcan et. Al has also proposed the great deluge based hyper heuristic with reinforcement learning for exam timetabling problem [4].

Finally, McMullan proposed an extended great deluge algorithm(EGD) for university course timetabling , which allows re-heating similar to simulated annealing, and found new best results for the 5 medium instances. 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)[16].Therefore, from these literature review, the success of the great deluge and its variants has been known.

3. Description of the Exam Timetabling Problem

The university exam timetabling problem can be defined the allocation of exams to a limited number of time periods in order to satisfy the chosen set of constraints. The size of the search space for a timetabling problem increases exponentially as the number of items to be scheduled increases and they are known to be NP-complete constraint optimization problems. In a more formal way, the timetabling literature defines two types of constraints. Hard Constraints are the constraints that must be satisfied at all times. Soft Constraints are not critical but their satisfaction is beneficial to students and/or the institution. Typically one cannot satisfy all soft constraints thus there is a need for a performance function measuring the degree of satisfaction of these constraints [1].The primary hard and soft constraints in exam timetabling problem can be find in [6]. Among them, we use the following four constraints and one soft constraint as shown in the table 1.

Table 1 Hard and Soft Constraint

HC1	No exams with common resources (e.g. students) assigned simultaneously.
HC2	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.)
HC3	Each examination must be assigned to a timeslot only for once.
HC4	All the examinations must be scheduled.
SC1	Spread conflict exams as even as possible or not in x consecutive timeslots or days. (i.e. A student should have at least a single timeslot in between his/her examinations in the same day).

The formalization of the problems can be shown in the following table:

Table 2 Problem Description

E	number of n exams : E_1, E_2, E_3… E_n
S	number of m students : S_1, S_2, S_3,S_m
T	number of k timeslots: T_1, T_2, T_3….T_k
C	$(c_{ij})_{n*n}$,the conflict matrix

In the conflict matrix, each element (denoted by c_{ij} where i, j …) is the numbers of students that have to be take both exams i and j. This is a symmetrical matrix of size n, where diagonal element c_{ii} 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 total cost of the following function exam timetable.

$$\frac{\sum w(|e_i - e_j|)c_{ij}}{S} \qquad (1)$$

In equation, | e_i -e_j | is the distance between the periods of each pair of examinations (e_i , e_j) with common students and cij is the number of students common to both examinations. It 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.

4. Proposed EGD based HH System

This section presents the general framework of the hyper heuristic and EGD based hyper heuristic system for making acceptance or not of the problem solution. The figure 1 represents the general concept of the hyper heuristic. In figure, the step 4 and 5 are the main stage of the hyper heuristic. The main contribution of this paper can be seen in step 5. As far as the author knows, EGD has been considered first time as move acceptance for hyper heuristic.

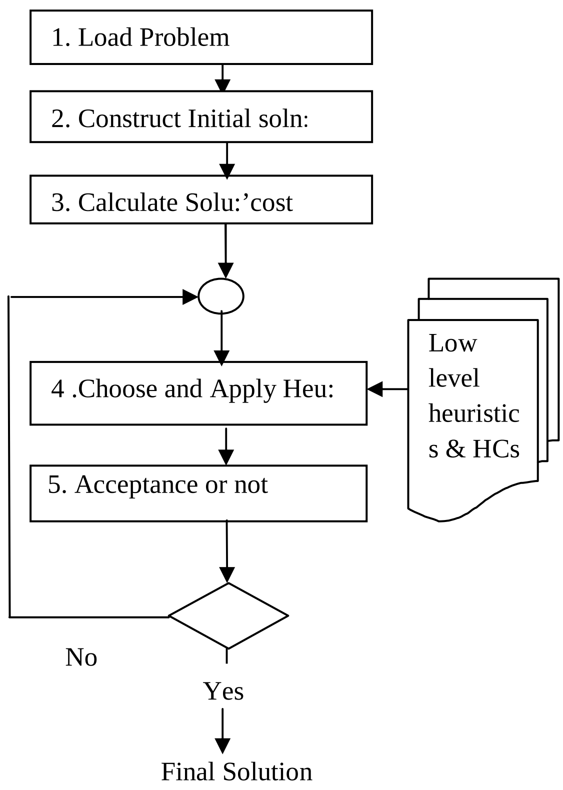

Figure 1. Overview of hyper heuristic

4.1 Dataset

The original version I, version II and the version IIc data files are all available at the site http://www.asap.cs.nott.ac.uk/resources/data.shtml. In each problem instance, the two files will be included. The first file shows the number of exam and enrolment of student for it. The second one displays the students list who takes which exams. The table 1 provides the characteristics of the Toronto dataset.

Table 3 Dataset Description

Instance	Exams	Enrolment	Density	Period	Capacity
Car91 I	682	56877	0.13	35	1550
Car92 I	543	55522	0.14	32	2000
Ear83 I	190	8109	0.27	24	350
Hecs92 I	81	10632	0.42	18	650
Kfu93	461	25118	0.06	20	1955
Ise 91	381	10918	0.06	18	635
Pur93 I	2419	120681	0.03	42	5000
Rye 92	486	45051	0.07	23	2055
Sta83I	139	5751	0.14	13	3024
Tre92	261	14901	0.18	23	655
Uta92	622	58979	0.13	35	2800
Ute92	184	11793	0.08	10	1240
Yor83I	181	6034	0.29	21	300

4.2. Initial solution

It is important to have an easy and quick way of generating an initial solution. It is to be feasible only by satisfying all hard constraints. Note that it is not necessary though that initial solution should be completely feasible. However, it is preferred to be as feasible as possible because the quality of initial solution would affect the final solution. Therefore, we use the largest enrolment (LE) heuristic to produce completely feasible solution in step 2 of the figure. It is the examination with the largest student enrolment is scheduled first It schedules first exam with the largest number of students. Then the cost of the initial solution is calculated by using the equation 1.

For the solution representation; there are two forms of assignment such as:

- Exam-Timeslot assignment and
- Exam-Classroom assignment.

Here, we use the first one as shown in the figure 2.

Timeslot	T_1	T_2	..	T_k
Exam	E_1,E_3	E_2	..	E_n

Figure 2. Representation of the solution

4.3 Low level Heuristic

To get the final optimized solutions, it also totally depends on the set of heuristics it can be choose from. Also, due to the performance changes of a number of heuristics over a search space, it is not easy to find a heuristic that always produces the best decisions. They are heuristics that allow movement through a solution space and that require domain knowledge and are problem dependent. Each heuristic creates its own heuristic search space that is part of the solution search space. There are many low level heuristics (LLH) in the literature, for example: mutational heuristic, ruin or recreate heuristic and so on. For low level heuristic module, in this paper, three types of heuristics are used. They are:

(1) Swapping timeslots with ensuring the feasibility
(2) Swapping two exams between different random timeslot with ensuring the feasibility.
(3) Adding or removing exam from the unscheduled list

4.4 Reinforcement Learning

For the heuristic selection process in hyper heuristics, machine learning techniques are vital to make the right choices. Learning can be achieved in an offline or online manner. An online learning hyper heuristic learns through the feedback obtained during the search process while solving a given problem. Moreover, it is better than offline method. Most of the existing online learning hyper heuristic incorporates reinforcement learning (RL). A simple P: 1(Additive adaption)-N: 1(Negative adaption) strategy and maximum utility method are used for RL and the utility values are allowed to change within an interval of [0, number of heuristics].It requires less a priori knowledge. Moreover, it is successfully applied to scheduling, control, game theory and so on but the quality of solution obtained by using is not satisfactory in many times. However, EGD can control to make a decision whether it is accepts or rejects, after applied the chosen heuristic to initial solution.

4.5 Extend Great Deluge

In fact, the concept of EGD algorithm is quite similar with the hyper heuristic method. EGD has advantages to require the tuning of a few input parameters that can represent the search time. It is extended to avoid the trap of local optima and to reduce the time required for an iterative search process. It can provide a wider test with the hidden data sets for consistency in the approaches. Moreover, it can be interesting to run all techniques at some future point with further hidden data sets. It is also proved to be both robust and general.

Because of these advantages, it has been successfully applied to many optimization problems such as buffer allocation problem, redundancy allocation problem and so on. Therefore, in this paper, it is investigated to make further improvement in hyper heuristic or not. The figure 3 shows the EGD based hyper heuristic system. It is implemented and tested on the Toronto exam timetabling benchmark data sets.

Firstly, the initial solution is constructed and then the initial cost, also known as the quality of the solution is calculated. To chooses the low level heuristic; reinforcement learning is used and applies the chosen heuristic to the initial solution. Then, the decision of acceptance of the current solution is checked by using the proposed EGD. At the step 5, we need to define the initial boundary level and the decay rate. Then, the current solution is accepted if the cost of current solution is less than or equal to the cost of initial solution. It employs reheating in order to relax the boundary condition to allow worse move to be applied to the current solution.

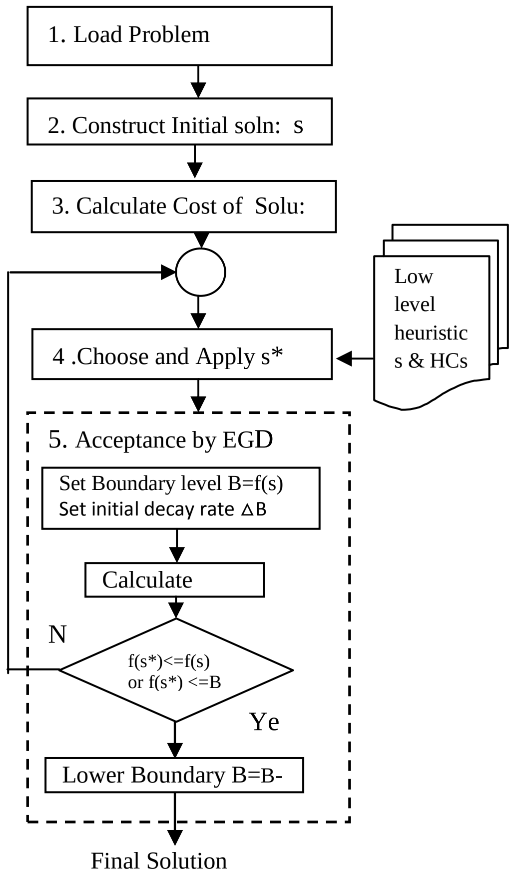

Figure.3. EGD based Hyper Heuristic System

5 Conclusions

Hyper heuristics are starting to prove themselves as fast and effective methods for solving complex real world optimization problems. Different performance of hyper heuristic can be achieved by different combination of heuristic selection methods and move acceptance criteria. Determining the best adaptation rate is a key issue of the reinforcement learning. However, the proposed extended great deluge based hyper heuristic can support to achieve the best performance results. The average best fitness reached is used as the performance criterion for all experiments. The performance is evaluated statistically using t-test. The results will be analyzed and reported in our next job.

References

[1] E. Burke, G. Kendall, E. Soubeiga. "A tabu-search hyperheuristic for timetabling and roistering", *Journal of Heuristic,* 9:451-470, 2003.

[2] E. Soubeiga. "Development and application of hyper heuristic to personnel scheduling." *PhD thesis, School of Computer Science, University on Nottingham, UK,* 2003.

[3] H. J. Obit, D. Landa-Silva,D. Quelhadj , M. Sevaux, " Non-Linear Great Deluge with learning Mechanism for Solving the Course Timetabling Problem", MIC-2009: The VIII Met heuristic International Conference.

[4] 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.

[5] E. K. Burke, Y. Bykov, "Solving Exam Timetabling Problems with Flex-Deluge Algorithm", PATAT 2006, pp. 370-372. ISBN 80-210-3726-1.

[6] 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, 12(1): 55-89, 2009.

[7] Carter M. and Laporte G.:" Recent Developments in Practical Exam Timetabling". In: *Burke E.K and Ross P.(eds.): (PATAT95), Lectures Notes in Computer Science* 1153, 3-21,1996.

[8] E. K. Burke, Kendall Graham and R. Ozcan, " Hyper Heuristics: A Survey of the State of the Art"

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

[10] Yuri Bykov, "Time-Predefined and Trajectory-Based Search: Single and Multiobjective Approaches to Exam Timetabling", 2003.