Journal of Theoretical and Applied Information Technology
20th June 2016. Vol.88. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

PARTIAL GRAPH HEURISTIC WITH HILL CLIMBING FOR EXAMINATION TIMETABLING PROBLEM

[1]ASHIS KUMAR MANDAL, [2]M N M KAHAR
[1,2]Faculty of Computer System and Software Engineering
University Malaysia Pahang (UMP), Pekan, Pahang, Malaysia
Email: [1]ashis.62@gmail.com,[1]ashis@hstu.ac.bd, [2]mnizam@ump.edu.my

ABSTRACT

Generating an examination timetable for an institution is a challenging and time-consuming task due to its inherent complexity and lots of constraints associated with scheduling the exams. In fact, it is a combinatorial optimization problem that requires heuristic searches to converge the solution into an optimized area. This paper presents combining the partial graph heuristic with hill climbing search (PGH-HC) to solve the examination timetabling problem. The approach first orders the exams using graph heuristics, and partially selected exams, which are generated based on predefined constant called exam assignment value (EAV), are assigned to timeslots and / or rooms. Next, the hill climbing search is employed to improve the partially scheduled exams. The above process continues until all exams have been scheduled. The overall aim here is to devise a straightforward approach which employs tuning of fewer parameters in solving the exam timetabling problem. Two benchmark examination timetabling datasets, Toronto un-capacitated datasets and ITC2007 capacitated datasets are considered for measuring the efficiency of the proposed method. We analyse the PGH-HC with different EAV, graph heuristics and termination criteria on these benchmark datasets. Experimental results reveal that, in general, PGH-HC with relatively a smaller EAV and a higher number of iterations (i.e. longer termination durations) is found to produce better quality solutions for all problem instances. Besides, the proposed approach is able to generate better quality solutions for all problem instances compared to traditional graph heuristic with hill climbing search (TGH-HC) and produce competitive results while comparing with the state-of-the-art approaches.

Keywords: *Combinatorial optimization problem, graph heuristics, hill climbing, scheduling, Timetabling*

1. INTRODUCTION

Solving the examination timetabling problem is a nontrivial task as it contains lots of constraints and these constraints conflict with each other. Most of the educational institutions provide considerable attention while scheduling their institutions' final examination. In algorithmic point of view, it is a NP-hard problem along with solving this timetabling problem with good quality solutions is rarely achievable using manual or even exact approach. Researchers have introduced various optimization techniques as seen in the scientific literature in order to solve the problem. Surveys provided by Carter, et al. [1] and Qu, et al. [2] on examination timetabling problem highlight numerous techniques, such as graph heuristic [3], tabu search [4], simulated annealing [5], great deluge[6], late acceptance hill climbing [7], evolutionary algorithms [8-10], constraint programming [11], case-based reasoning [12] and fuzzy methodologies [13], for solving this issue. Besides, PATAT series of conference proceedings, which were held from 1995 to 2014, have given a good description of the problem and various techniques (available at the following link http://www.patatconference.org/).

Examination timetabling problem is defined as assigning a set of exams into a set of finite timeslots and rooms and at the same time satisfying some constraints. These constraints are categorized as hard constraints and soft constraints. The hard constraints must be satisfied for obtaining a feasible solution, whereas the quality of the solution depends on soft constraints satisfaction. Soft constraints can be violated but a penalty value is given for every violation. A typical hard constraint might be no student takes more than one exam at the same time and a soft constraint might be to spread the exams evenly over the exam periods. Examination timetabling problem is classified as

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

capacitated and un-capacitated. In the un-capacitated timetabling problem, room capacity is not considered but room capacity is considered as hard constraint for capacitated timetabling problem.

Most of the earlier works attempt to generate examination timetabling in two phases: firstly, to construct one or more complete initial feasible solution(s), and then to improve the quality of the feasible solution(s) [14-16]. Usually, sequential graph heuristics are employed in constructing the initial solution due to their simplicities and capabilities in generating feasible solutions. In improvement phase, various local search and population-based meta-heuristics such as hill climbing, simulated annealing, great deluge algorithm, genetic algorithm and many others are used to improve the solution quality. Usually, in this sequential approach, the initial solution biases the result of the improvement steps, with a good initial solution tending to produce good improved solution [14, 17]. Additionally, if the initial solution is bad (local optima), sometimes the improvement phase is unable to work effectively in producing quality solution [18, 19].

In this work, we present a partial exam assignment with graph heuristic and hill climbing approach (PGH-HC) to solve the examination timetabling problem. The procedure starts with ordering the exams using graph heuristic and then the ordered exams are selected for scheduling based on a predefined constant. This constant refers to as exam assignment value (EAV). Then, these partially scheduled exams are improved using hill climbing method. The process repeats until all exams have been scheduled. In any case, where an exam is unable to be scheduled, we implement '*complexExamManager'* mechanism during the exam scheduling process.

The paper is organized as follows: section 2 presents the examination timetabling problems and their formulations, whereas section 3 describes the partial graph heuristic with hill climbing approach (PGH-HC) in details. Section 4 highlights traditional graph heuristic with hill climbing approach (TGH-HC). In section 5, experimental setups are presented follow with results and discussions in section 6. Finally, section 7 concludes the research works executed in this paper.

2. EXAMINATION TIMETABLING PROBLEM DESCRIPTIONS AND FORMULATIONS

In this paper, two distinct datasets, namely capacitated (ITC2007) and un-capacitated (Toronto) datasets are used for testing. These datasets are described below:

2.1 Un-capacitated Benchmark Datasets

In 1996, Carter and Laporte introduced the Toronto examination benchmark datasets. These datasets are un-capacitated examination timetabling datasets where unlimited number of seats during exam assignment is assumed. The Toronto datasets consist of 13 problem instances. Table 1 shows the properties of the datasets. The datasets are available at http://www.asap.cs.nott.ac.uk/resources/data.shtml.

Table 1: Toronto Datasets

Datasets	Number of timeslots	Number of Exams	Number of Students	Conflict Density
car-s-91	35	682	16925	0.13
car-f-92	32	543	18419	0.14
ear-f-83	24	190	1125	0.27
hec-s-92	18	81	2823	0.42
kfu-s-93	20	461	5349	0.06
lse-f-91	18	381	2726	0.06
pur-s-93	42	2419	30029	0.03
rye-s-93	23	486	11483	0.07
sta-f-83	13	139	611	0.14
tre-s-92	23	261	4360	0.18
uta-s-92	35	622	21267	0.13
ute-s-92	10	184	2750	0.08
yor-f-83	21	181	941	0.29

In Toronto datasets, one hard and one soft constraint are considered. Hard constraint is that no students are allowed to sit two or more exams simultaneously (also known as clashing constraint). In addition, soft constraint is to spread exams evenly so that students get ample time for last minute preparation before the next exams. Eq-1 is an objective or penalty function of Toronto datasets and the aim of the function is to reduce the soft constraint violation as much as possible providing hard constraint is satisfied. From this equation, proximity cost 16 is defined when exams are assigned successively. With the increase of gap between exams, proximity cost is reduced. Therefore, penalty value 8, 4, 2, and 1 are assigned for 1, 2, 3, and 4 time slot gap between exams respectively. More details description of the equation is found in [1].

Journal of Theoretical and Applied Information Technology
20th June 2016. Vol.88. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$\min \frac{\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} c_{ij} * proximity(t_i, t_j)}{M} \quad \text{(Eq-1)}$$

Where

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

N is the number of examinations.

M is the total number of students.

T is the number of available timeslots.

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∈{1,..., N}.

t_k ($1 \le t_k \le T$) specifies the assigned timeslot for examination k (k∈{1,..., N}).

2.2 Capacitated benchmark datasets

The ITC2007 examination datasets, which contain real-world hard and soft constraints, are capacitated problem. Unlike un-capacitated datasets, they are more complex to solve. Eight instances of the datasets are publicly available (see Table 2). In Table 2, A1 is the number of registered students, A2 indicates number of exams, A3 represent the number of timeslots, A4 indicates room availability, A5 means the number of period related hard constraints, A6 is the number of hard constraints associated with rooms and A7 is the conflict density

Table 2: ITC2007 Examination Datasets

Datasets	A1	A2	A3	A4	A5	A6	A7
Exam1	7,833	607	54	7	12	0	5.05%
Exam2	12,484	870	40	49	12	2	1.17%
Exam3	16,365	934	36	48	170	15	2.62%
Exam4	4,421	273	21	1	40	0	15.0%
Exam5	8,719	1018	42	3	27	0	0.87%
Exam6	7,909	242	16	8	23	0	6.16%
Exam7	13,795	1096	80	15	28	0	1.93%
Exam8	7,718	598	80	8	20	1	4.55%

Feasibility of the datasets is accomplished when all exams are assigned into timeslots and rooms without violation of the hard constraints. The hard constraints for ITC2007 examination datasets are defined as follows:

H1. Any student cannot sit more than one exam at the same time.

H2. The exam capacity should not exceed the room capacity.

H3. The exam length should not violate the period length.

H4. Three ordering of exams must be respected.
- Precedences: exam i will be scheduled before exam j.
- Exclusions: exam i and exam j must not be scheduled at the same period.
- Coincidences: exam i and exam j must be scheduled at the same period.

H5. Room exclusiveness must be maintained. For example, exam i must take place only in room number 206.

The more the soft constraints are satisfied, the better the quality of solutions is obtained. Soft constraints for the ITC2007 examination datasets are defined as follows:

S1. *Two Exams in a Row(C_s^{2R}):* Avoid the number of occasions where a student sits consecutive exams on the same day.

S2. *Two Exams in a Day (C_s^{2D}):* Avoid the number of occasions where a student sits two exams in a day. Note that when exams are one after another, this is counted as Two Exams in a Row for avoiding duplication.

S3. *Spreading of Exams (C_s^{PS}):* Exams should be spread as evenly as possible over the time periods.

S4. *Mixed Duration (C^{NMD}):* Avoid the number of occasions where exams with different durations are scheduled into the same room.

S5. *Scheduling of Larger Exams (C^{FL}):* Avoid the number of occasions where the largest exams are assigned later in the timetable.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

S6. *Room Penalty(* C^R *):* Avoid the number of occasions where certain rooms with associated penalty are used for scheduling

S7. *Period Penalty (* C^P *):* Avoid the number of occasions where certain periods with associated penalty are used for scheduling

The objective function is formularized as in Eq-2 [20]. It attempts to minimize the violation of (penalty) soft constraints for producing good quality solution without violating the hard constraints.

$$\min \sum_{s \in S} (W^{2R} C_s^{2R} + W^{2D} C_s^{2D} + W^{PS} C_s^{PS}) + W^{NMD} C^{NMD} + W^{FL} C^{FL} + C^R + C^P \quad \text{(Eq-2)}$$

In this equation, W (with different subscription) stands for related weight for each of the soft constraints, whereas S indicates a set of students. Associated weighted values are not multiplied with C^R and C^P as these values are already added in the definition. Weights of the datasets are presented in Table 3. Details explanation of the ITC2007 exam tracks as well as the mathematical model will be found in [20, 21].

Table 3: Weights of the ITC2007 Examination Datasets

Datasets	weight for two in a day (W^{2D})	weight for two in a row (W^{2R})	weight for period spread (W^{PS})	weight for no mixed duration (W^{NMD})	weight for the front load penalty (W^{FL})
Exam1	5	7	5	10	5
Exam2	5	15	1	25	5
Exam3	10	15	4	20	10
Exam4	5	9	2	10	5
Exam5	15	40	5	0	10
Exam6	5	20	20	25	15
Exam7	5	25	10	15	10
Exam8	0	150	15	25	5

3. PARTIAL GRAPH HEURISTIC WITH HILL CLIMBING (PGH-HC) APPROACH

3.1 Graph Heuristic Selection and Ordering

The basic examination timetabling problem can be classified as graph colouring problem where a set of exams represents a set of vertices and the confliction of exams (hard constraint) can be represented as edges. Vertices with common edges have different colours, indicating timeslots. Usually, in timetabling problem, a vast number of exams have to be accommodated into limited number of timeslots, which is a challenging task. This task could be smoother if they are ordered based on difficulty for scheduling. Graph heuristics are such type of ordering strategies where most difficult exams are chosen for scheduling first. The ordering is accomplished with different heuristics. However, scientific literature frequently uses largest degree (LD), largest enrolment (LE), largest weighted degree (LWD) and saturation degree (SD) heuristics for ordering. The heuristics are described as follows.

- Largest degree (LD): This technique orders the exams based on the largest number of conflicting examinations.
- Largest weighted degree (LWD): This heuristic is similar to the largest degree except the exams are ordered based on the number of students in conflict.
- Largest enrolment (LE): The exams are ordered based on the number of registered students in the exams.
- Saturation degree (SD): The exams are ordered based on the number of remaining timeslots available; exams with the least number of available timeslots in the timetable are given priority to be scheduled first. SD is a dynamic heuristic where the ordering of exams is updated as the exams being scheduled.

In SD, we use three (3) ordering strategies which include SD(LD), SD(LWD) and SD(LE). For example in SD(LD), the exams are ordered according to LD and during exam allocation, these exams are scheduled based on SD. If a condition where the number of remaining timeslots is similar between exams, these exams are selected based on LD feature. The same approach is employed for SD(LWD) and SD(LE).

3.2 Description of the Model

The basic framework of our proposed approach is illustrated in Figure 1. Initially, a heuristic, H was chosen as listed in section 3.1 (line 1). Next, we set the exam assignment value (EAV) to designate the number of exams for scheduling (line 2). In our experiment, the EAV is represented as a percentage of total exams. For instance, if EAV=25% with total exams of 165, only 41 exams will be selected for scheduling (we refer to this as partial exams).

The iteration starts with ordering the un-scheduling exams using the selected heuristic (line

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4). Afterwards, these partial exams are selected based on value EAV (line 5) from the ordered exams. These partial exams are then randomly scheduled to available timeslots and rooms (i.e. for capacitated dataset, ITC2007) without violating the hard constraints (line 6). During scheduling, if an exam is unable to be assigned to any timeslots, a *complexExamManager* is used which attempt to handle those difficulties to schedule exam (line 7). A details explanation of the *complexExamManager* is discussed in section 3.3. When all partial exams are scheduled successfully, they are removed from the unscheduled exam set and the penalty cost is calculated (refer Eq-1 for Toronto and Eq-2 for ITC2007) based on the partially scheduled exams so far (lines 9-11). Next, hill climbing (HC) is used to improve the quality of the partially scheduled exams with the aim of satisfying the soft constraints (i.e. minimizing the penalty value) – line 13. The above process repeats for the next batch of unscheduled partial exams. These exams are reordered again. Once these partial exams are scheduled, they are then improved using hill climbing method. The entire process repeats until all exams have been scheduled. Finally, the final solution and its penalty are stored.

```
1. Choose a heuristic H from [SD(LE),SD(LWD),SD(LE)]
2. Set exam assignment value, EAV
3.  while Until end of all exams assigned to timeslots
4.     Order the unscheduled exams using heuristic H and
       put into an unscheduled set
5.     Select partial exams from the ordered list based on EAV
6.     Schedule partial exams while satisfying all hard
       constraints
7.     If An exam is unable to schedule
           Use complexExamManager
8.     end if
9.     If Current partial exams are scheduled successfully
10.          Remove them from unscheduled set and insert into
             a scheduled set
11.         Calculate (temporary) penalty cost of all scheduled
             exams so far
12.    end If
13.    Use Hill Climbing to improve the penalty value
14.  end while
15. Return final penalty cost as result
```

Figure 1: Partial Graph Heuristics with Hill Climbing (PGH-HC) Approach

3.3 *ComplexExamManager* Procedure

It is observed that when a dataset has more conflict density and multiple hard constraints, exams tend to face difficulties during the scheduling. In the partial exam scheduling, there might be a condition where exams are unable to be assigned to any timeslots or rooms. Additionally, in partial exam scheduling, the unscheduled exams have to be scheduled into an improved partial scheduling solution (assuming a set of partial exams has been completely scheduled and improved, and now we are in the second round of partial exam scheduling). Therefore, the unscheduled exams get less freedom to be scheduled into timeslots and rooms. Hence, when an exam cannot be scheduled to any timeslots and rooms, then a mechanism called *complexExamManager* is employed so that the exam can be scheduled. Firstly, when a particular exam cannot be scheduled, it is considered as '*complexExam'*. Then a particular timeslot and/or room is selected which has less conflicted exams with the *complexExam*. The conflicting exams are then moved to different timeslots and/or rooms while maintaining the feasibility of the current solution. Next, the '*complexExam*' are assigned to the selected timeslot and/or available room. Otherwise, the next less-conflicted timeslot are taken and the same process is executed until all available timeslots are checked. However, if this step still unable to schedule the exam, we indicate that the exam is very complex exam. We then attempt to schedule the exam in the next iteration. Figure 2 shows the algorithm for scheduling the complex exam.

```
Select the unscheduled exam. we call it complexExam
Select the current partial scheduling vector
for each Timeslot
   Count the number of conflicted exams with
   complexExam
end for
Sort all the timeslots in ascending order according to
 number of conflicts and insert into a queue Q
while there exist timeslot in Q
   De-queue timeslot T
   Try to schedule conflicted exams in another time
   slot(s)
   if All conflicted exams are scheduled
         Try to move complexExam in the timeslot T
         and any available room in T
          if complexExam is scheduled successfully
                Update partial scheduling vector
                  Return from the loop
          end if
   end if
end while
```

Figure 2: ComplexExamManager Procedure

3.4 Improvement using Hill Climbing (HC)

In the next step, hill climbing (HC) is employed to improve the quality of the partially of scheduled exams. That is, it minimizes the penalty cost. The selection of the hill climbing is preferred due to the fact that it does not need to set up any parameter settings as well as it can exploit the solution space quickly.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Figure 3 illustrates the improvement using hill climbing approach. The algorithm starts with initializing the partial solution vector as current solution and the cost is derived from previous partial scheduling. Number of neighbourhoods is defined, which indicates the number of different solutions after twigging the initial solution vector. Stopping criteria are defined by the number of iterations or time duration, indicating how many times the improvement of solution is occurred. During the iteration, different neighbourhood structures are used for generating candidate solutions and a promising candidate solution s^* is identified. Then the cost of candidate solution $f(s^*)$ is compared with the cost of current solution $f(s)$. If $f(s^*)$ is equal or less than $f(s)$, the current solution s is replaced by the candidate solution s^*. This iteration end until meeting the stopping criteria and finally the partial best solution is returned.

```
Set the initial partial solution vector s as current solution
Set the initial cost function f(s) as current cost function
Set n - number of neighborhood structures
Set a time duration or number of iterations as stopping criteria
while stopping criteria do not meet do
   Calculate neighbor solutions by applying neighborhood
   structures (N1, N2, N3,...., Nn) and consider best
   solution as candidate solution
   if f(s*) ≤ f(s)
      s = s*
   end if
end while
```

Figure 3: Hill Climbing Procedure (called by Figure 1, line 13)

4. TRADITIONAL GRAPH HEURISTIC WITH HILL CLIMBING APPROACH (TGH-HC)

TGH-HC is a two-step approach for solving examination timetable. At first, a complete feasible solution is constructed and then hill climbing search improves the quality of the solution. Figure 4 illustrates the approach. Exams are ordered using six (6) graph heuristics: LD, LWD, SE, SD(LD),SD(LWD), SD(LE). For each heuristic ordering, initial feasible solution and its penalty are computed for thirty (30) individual runs. Among these penalty values, the best value with corresponding solution vector is taken as an initial solution. Finally, hill climbing improves the solution vector further and returns final penalty cost if its solution vector satisfy all hard constraints. We will also implement the TGH-HC onto the experimented examination datasets to allow for comparison with the proposed method.

```
do initial ordering based on heuristics H [LD,LWD,LE,
SD(LE),SD(LWD),SD(LE)]
for each Heuristic of H
  for 30 iterations
     Construct initial feasible solution
     Calculate initial feasible penalty
  end for
end for
Select best penalty cost and solution vector
Use Hill climbing (HC) for improvement
if Final solution vector satisfies all hard constraints
    Return final penalty cost as result
end if
```

Figure 4: Traditional Graph Heuristic with Hill Climbing (TGH-HC) Approach

5. EXPERIMENTAL SETUP

In order to test the PGH-HC approach, the Toronto datasets as well as the ITC 2007 exam datasets are used.

For Toronto benchmark datasets, we choose twelve instances of the datasets (Table 1). Four different EAV, including 10%, 25%, 50% and 75% from the total number of exams, are used for experiment. Moreover, in the partial construction of exams three different graph heuristics including SD(LD), SD(LWD), SD(LE) are employed. For improvement of partial solution quality with hill climbing (HC), three different termination criteria, which include 10,000, 50,000 and 100,000 iterations, are set as stopping criteria. Three neighbourhood structures which are introduced in hill climbing search are described as follows:

- ***N*$_1$**: randomly select an exam and move the selected exam to a randomly selected timeslot.
- ***N*$_2$**: randomly select two exams and swap their timeslots.
- ***N*$_3$**: select two timeslots randomly and move all exams between the two timeslots.

In each iteration of the improvement phase, only one neighbourhood structure that provides the best penalty cost among three is selected as candidate neighbourhood. Note that, same stopping criteria and neighbourhood structures are applicable for TGH-HC in solving Toronto datasets.

We experiment with eight instances from the ITC2007 examination track (Table 2). Here we

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

incorporate two EAV, which includes 5% and 10% from the total number of exams. Moreover, like Toronto, we use three graph heuristics comprising SD(LD), SD(LWD), SD(LE) in partial construction phase. However, for ITC2007, we do not use predefined iteration as termination criterion. Rather, we use time duration in such a way that it defines iteration dynamically for each partial improvement. This time duration defines how long the program runs for complete solution generation and improvement. We experiment with two time durations: 600 seconds and 3600 seconds. We set the termination criterion for 600 seconds as this was used during the ITC2007 second exam timetable competition. Following neighborhood heuristics are assigned for improvement phase:

- ***N_1***: Move a random exam to a different timeslot and room.
- ***N_2***: Select two exams randomly and swap their timeslots and rooms.
- ***N_3***: Move an exam to a different room within the same timeslot.
- ***N_4:*** Move two random exams to different timeslots and rooms.

In each iteration during improvement, only one neighborhood is selected randomly and employed if the solution it provides is feasible; otherwise, we choose different neighborhood. Note that, same time durations and neighbourhood structures are applicable for TGH-HC in solving ITC2007 datasets.

For both Toronto and ITC2007 exam track, each instance was experimented with different combination of EAV, graph heuristics and iterations (time durations for ITC2007). We ran each experiment 30 times using different random seeds. Finally, the programs were implemented in Java (Java SE 7) and performed on Intel Corei3 (3 GHz) PCs with 2 GB RAM running Windows 7 Professional SP3.

6. RESULTS AND DISCUSSIONS

In our experiments, firstly, TGH-HC and PGH-HC procedures are applied on Toronto datasets. Next, these two procedures are used on ITC2007 exam track. Different EAV, graph heuristics, and termination criteria are considered during the experiments as described in the experimental setup.

We have performed t-test for demonstrating whether proposed PGH-HC performs better than TGH-HC in solving both benchmark datasets. Confidence interval is set at 95% (α=0.05 level of significance). The null hypothesis (H_0) is defined as there is no significant difference between TGH-HC and PGH-HC. On the other hand, alternative hypothesis (H_1) is defined as PGH-HC is better than TGH-HC. The target is to validate the H_1.

6.1 Experimental Results with Toronto Datasets

Table 4(a)-(d) illustrate the effect of the penalty values of Toronto datasets where three different iterations and four EAV are considered. Best and average penalty costs for all datasets are enlisted where both TGH-HC and PGH-HC approaches are used. It is observed that for the same EAV, with a large number of iterations, the penalty costs are usually smaller for both TGH-HC and PGH-HC approach. However, with the same number of iterations, an increase in the EAV value tends to produce larger penalty costs (poorer quality of the solution) for PGH-HC approach, but penalty costs are unchanged for TGH-HC approach because it is independent of EAV. It is observed that good quality solutions are obtained when EAV is small (i.e. 10%) and the iteration number is high (i.e. 100,000). It is also noticed that in general PGH-HC produces better results compared to TGH-HC for the same EAV value and iterations. During smaller EAV, the differences of penalty values between these two approaches are more noticeable than higher EAV. For example, with 75% EAV, we do not get significant difference in penalty values between the two approaches, whereas opposite of that is true for smaller EAV, such as 10%.

Table 5 shows the comparison between TGH-HC and PGH-HC where best penalty values (best costs with corresponding average values) in the experiments are highlighted. Best results are taken from Table 4(a)-(b) (results with bold font).

Referring to both Table 5 and Table 7, the arrangement of the column are as follow: column A contains datasets. Column B represents initial best solution and graph heuristic that produces the best solution. Column C and D represent best and average costs after hill climbing improvement is employed respectively. For PGH-HC, best and average costs are represented in column E and F respectively, whereas column G and H show the graph heuristic and EAV that produce the best costs respectively. Finally, column I shows the improvement of PGH-HC over TGH-HC in percentage. Finally, column J presents the *p*-values from t-test between these two approaches.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 4: Penalty Values of Toronto Datasets for Different Iterations

(a) with 10% EAV

Datasets	10,000 iterations				50,000 iterations				100,000 iterations			
	TGH-HC		PGH-HC		TGH-HC		PGH-HC		TGH -HC		PGH-HC	
	Best	Avg	Best	Avg	Best	Avg	Best	Avg	Best	Avg	Best	Avg
car-s-91	5.60	5.81	5.15	5.33	5.54	5.74	**5.08**	5.28	**5.54**	5.74	5.09	5.27
car-f-92	4.75	4.94	4.36	4.49	4.72	4.92	**4.23**	4.48	**4.71**	4.92	4.28	4.48
ear-f-83	40.50	41.95	37.46	40.14	40.48	41.68	**37.06**	40.13	**39.33**	41.75	37.93	40.24
hec-s-92	12.08	12.65	11.21	11.80	12.06	12.48	11.19	11.85	**11.91**	12.56	**10.98**	11.76
kfu-s-93	16.05	16.81	14.72	15.76	16.03	16.77	**14.53**	15.70	**16.03**	16.77	14.90	15.68
lse-f-91	12.43	13.12	11.13	11.94	12.29	13.03	11.21	11.90	**12.13**	13.03	**11.12**	11.95
rye-s-93	10.44	10.85	9.42	10.00	10.38	10.80	9.45	9.89	**10.38**	10.80	**9.40**	9.88
sta-f-83	157.54	158.41	157.39	158.38	157.54	158.40	**157.23**	158.34	**157.54**	158.39	157.31	158.43
tre-s-92	9.56	9.84	8.52	8.93	9.54	9.82	8.59	8.88	**9.55**	9.81	**8.55**	8.89
uta-s-92	3.88	3.97	3.50	3.61	3.86	3.98	**3.43**	3.59	**3.83**	3.93	3.44	3.59
ute-s-92	28.94	30.16	26.64	28.52	28.94	30.15	**26.57**	28.39	**28.94**	30.15	26.99	28.32
yor-f-83	40.51	42.04	38.88	41.62	40.46	41.27	**38.83**	41.20	**40.20**	41.26	39.71	41.82

(b) with 25% EAV

Datasets	10,000 iterations				50,000 iterations				100,000 iterations			
	TGH-HC		PGH-HC		TGH-HC		PGH-HC		TGH -HC		PGH-HC	
	Best	Avg	Best	Avg	Best	Avg	Best	Avg	Best	Avg	Best	Avg
car-s-91	5.60	5.81	5.21	5.58	5.54	5.74	5.21	5.54	5.54	5.74	5.29	5.52
car-f-92	4.75	4.94	4.39	4.66	4.72	4.92	4.46	4.64	4.71	4.92	4.34	4.62
ear-f-83	40.50	41.95	37.86	40.79	40.48	41.68	38.98	40.86	39.33	41.75	38.31	40.73
hec-s-92	12.08	12.65	11.28	12.03	12.06	12.48	11.27	11.93	11.91	12.56	11.02	11.95
kfu-s-93	16.05	16.81	14.84	15.96	16.03	16.77	15.06	15.95	16.03	16.77	14.91	15.97
lse-f-91	12.43	13.12	11.65	12.46	12.29	13.03	11.66	12.51	12.13	13.03	11.58	12.50
rye-s-93	10.44	10.85	9.63	10.26	10.38	10.80	9.52	10.22	10.38	10.80	9.61	10.09
sta-f-83	157.54	158.41	157.39	158.38	157.54	158.40	157.43	158.37	157.54	158.39	157.31	158.43
tre-s-92	9.56	9.84	8.80	9.22	9.54	9.82	8.74	9.20	9.55	9.81	8.86	9.18
uta-s-92	3.88	3.97	3.64	3.73	3.86	3.98	3.54	3.71	3.83	3.93	3.56	3.70
ute-s-92	28.94	30.16	27.16	28.93	28.94	30.15	26.80	29.11	28.94	30.15	27.34	28.99
yor-f-83	40.51	42.04	39.64	42.12	40.46	41.27	39.84	41.72	40.20	41.26	38.92	41.79

(c) with 50% EAV

Datasets	10,000 iterations				50,000 iterations				100,000 iterations			
	TGH-HC		PGH-HC		TGH-HC		PGH-HC		TGH -HC		PGH-HC	
	Best	Avg	Best	Avg	Best	Avg	Best	Avg	Best	Avg	Best	Avg
car-s-91	5.60	5.81	5.58	5.83	5.54	5.74	5.53	5.85	5.54	5.74	5.58	5.80
car-f-92	4.75	4.94	4.62	4.85	4.72	4.92	4.58	4.81	4.71	4.92	4.60	4.80
ear-f-83	40.50	41.95	38.31	41.25	40.48	41.68	38.94	41.26	39.33	41.75	37.96	41.26
hec-s-92	12.08	12.65	11.43	12.24	12.06	12.48	11.27	12.12	11.91	12.56	11.22	12.18
kfu-s-93	16.05	16.81	15.12	16.30	16.03	16.77	15.03	16.26	16.03	16.77	15.16	16.38
lse-f-91	12.43	13.12	11.99	12.85	12.29	13.03	11.89	12.84	12.13	13.03	12.00	12.80
rye-s-93	10.44	10.85	9.78	10.50	10.38	10.80	9.78	10.48	10.38	10.80	9.84	10.63
sta-f-83	157.54	158.41	157.45	158.44	157.54	158.40	157.37	158.36	157.54	158.39	157.71	158.44
tre-s-92	9.56	9.84	8.78	9.51	9.54	9.82	8.96	9.46	9.55	9.81	8.98	9.47
uta-s-92	3.88	3.97	3.70	3.86	3.86	3.98	3.64	3.85	3.83	3.93	3.66	3.85
ute-s-92	28.94	30.16	27.03	29.30	28.94	30.15	27.02	29.31	28.94	30.15	27.67	29.31
yor-f-83	40.51	42.04	40.65	42.30	40.46	41.27	39.34	42.29	40.20	41.26	39.71	42.22

(d) with 75% EAV

Datasets	10,000 iterations				50,000 iterations				100,000 iterations			
	TGH-HC		PGH-HC		TGH-HC		PGH-HC		TGH -HC		PGH-HC	
	Best	Avg	Best	Avg	Best	Avg	Best	Avg	Best	Avg	Best	Avg
car-s-91	5.60	5.81	5.68	5.99	5.54	5.74	5.68	5.91	5.54	5.74	5.67	5.96
car-f-92	4.75	4.94	4.63	4.95	4.72	4.92	4.57	4.93	4.71	4.92	4.66	4.94
ear-f-83	40.50	41.95	38.58	41.41	40.48	41.68	38.76	41.22	39.33	41.75	38.30	41.54
hec-s-92	12.08	12.65	11.47	12.16	12.06	12.48	11.22	12.03	11.91	12.56	10.99	12.01
kfu-s-93	16.05	16.81	15.60	16.83	16.03	16.77	15.59	16.79	16.03	16.77	15.58	16.78
lse-f-91	12.43	13.12	12.17	13.07	12.29	13.03	12.23	13.06	12.13	13.03	12.08	13.05
rye-s-93	10.44	10.85	10.05	10.82	10.38	10.80	10.01	10.62	10.38	10.80	10.03	10.61
sta-f-83	157.54	158.41	157.45	158.52	157.54	158.40	157.35	158.36	157.54	158.39	157.53	158.29
tre-s-92	9.56	9.84	9.11	9.58	9.54	9.82	9.02	9.57	9.55	9.81	9.09	9.57
uta-s-92	3.88	3.97	3.75	3.94	3.86	3.98	3.74	3.92	3.83	3.93	3.75	3.93
ute-s-92	28.94	30.16	27.78	29.67	28.94	30.15	27.61	29.84	28.94	30.15	27.69	29.67
yor-f-83	40.51	42.04	39.78	42.29	40.46	41.27	40.32	42.57	40.20	41.26	39.46	42.06

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 5: Comparison of Result obtained by TGH- HC and PGH-HC for Toronto Datasets

Datasets (A)	TGH-HC: Initial Solution with Graph Heuristic Ordering (B)	TGH-HC: Hill Climbing Best (C)	TGH-HC: Hill Climbing Avg (D)	PGH-HC: Best (E)	PGH-HC: Avg (F)	PGH-HC: Graph Heuristic Ordering (G)	PGH-HC: EAV (%) (H)	Improvement % = $\frac{\lvert E-C \rvert}{C}\times 100$ (I)	t-test *p*-value (J)
car-s-91	8.33 – LD	5.54	5.74	5.08	5.28	SD(LWD)	10%	8.30	2.28E-25
car-f-92	7.00 – LD	4.71	4.92	4.23	4.48	SD(LE)	10%	10.19	1.33E-24
ear-f-83	52.35 - SD(LE)	39.33	41.75	37.06	40.13	SD(LE)	10%	5.77	3.76E-07
hec-s-92	16.21 - SD(LWD)	11.91	12.56	10.98	11.76	SD(LD)	10%	7.81	3.01E-12
kfu-s-93	23.68 - (LD)	16.03	16.77	14.53	15.70	SD(LD)	10%	9.36	1.06E-12
lse-f-91	18.83 - (LE)	12.13	13.03	11.12	11.95	SD(LE)	10%	8.33	5.08E-15
rye-s-93	18.28 - SD(LD)	10.38	10.80	9.40	9.88	SD(LWD)	10%	9.44	1.14E-17
sta-f-83	166.43 - SD(LE)	157.54	158.39	157.23	158.34	SD(LE)	10%	0.20	0.034875
tre-s-92	12.07- SD(LE)	9.55	9.81	8.55	8.89	SD(LWD)	10%	10.47	3.84E-28
uta-s-92	5.53 – LE	3.83	3.93	3.43	3.59	SD(LE)	10%	10.44	1.63E-31
ute-s-92	38.03 – SD(LD)	28.94	30.15	26.57	28.39	SD(LWD)	10%	8.19	1.06E-13
yor-f-83	49.80 – LD	40.20	41.26	38.83	41.20	SD(LE)	10%	3.41	0.0006

Table 6: Penalty Values of ITC2007 Datasets for Different Termination Criteria

(a) with 5% EAV

Datasets	600 seconds				3600 seconds			
	TGH-HC		PGH-HC		TGH-HC		PGH-HC	
	Best	Avg	Best	Avg	Best	Avg	Best	Avg
Exam1	12,421	13,048.88	7,961	8,375.847	**10,648**	10,972.4	**6,377**	6,997.27
Exam2	2,807	3,766.787	1,716	2,535.1	**1,545**	2,189.142	**760**	960.30
Exam3	43,098	47,160.2	19,963	24,573.4	**23,257**	27,476.22	**15,092**	16,893.63
Exam4	34,241	34,937.07	23,000	26,226.28	**33,648**	34,355.99	**21,676**	27,825.62
Exam5	15,643	16,773.46	9,724	11,379.87	**7,955**	8,517.41	**5,224**	6,153.29
Exam6	29,630	33,880.08	26,170	28,354.36	**29,515**	30,703.08	**26,165**	27,519.42
Exam 7	19,080	21,612.42	12,007	13,647.47	**12,568**	13,745.5	7,219	7,772.40
Exam 8	23,315	25,002.29	12,482	16,639.03	**13,029**	15,487.32	10,520	16,274.30

(b) with 10% EAV

Datasets	600 seconds				3600 seconds			
	TGH-HC		PGH-HC		TGH-HC		PGH-HC	
	Best	Avg	Best	Avg	Best	Avg	Best	Avg
Exam1	12,421	13,048.88	8,027	9,218.47	10,648	10,972.4	6,942	7,602.8
Exam2	2,807	3,766.787	1,751	3,783.33	1,545	2,189.142	816	1,007.8
Exam3	43,098	47,160.2	21,435	26,281.83	23,257	27,476.22	16,057	18,440.37
Exam4	34,241	34,937.07	22,211	27,380.74	33,648	34,355.99	22,408	25,636.95
Exam5	15,643	16,773.46	10,039	12,172.8	7,955	8,517.41	5,944	7,379.4
Exam6	29,630	33,880.08	27,005	29,456.61	29,515	30,703.08	26,630	28,201.38
Exam7	19,080	21,612.42	13,318	14,455.87	12,568	13,745.5	**7,201**	7,453.69
Exam8	23,315	25,002.29	12,635	17,945.13	13,029	15,487.32	**10,201**	13,980.27

Table 7: Comparison of Result obtained by TGH- HC and PGH-HC for ITC2007 Exam Datasets

Datasets (A)	TGH-HC: Initial Solution with Graph Heuristic Ordering (B)	TGH-HC: Hill Climbing Best (C)	TGH-HC: Hill Climbing Avg (D)	PGH-HC: Best (E)	PGH-HC: Avg (F)	PGH-HC: Graph Heuristic Ordering (G)	PGH-HC: EAV (%) (H)	Improvement % = $\frac{\lvert E-C \rvert}{C}\times 100$ (I)	t-test p-value (J)
Exam1	25,989-SD(LE)	10,648	10,972.40	6,377	6,997.27	SD(LE)	5%	40.11	4.81E-46
Exam2	30,960- SD(LE)	1,545	2,189.14	760	960.30	SD(LE)	5%	50.81	6.8E-19
Exam3	85,356-SD(LD)	23,257	27,476.22	15,092	16,893.63	SD(LE)	5%	35.11	1.21E-27
Exam4	41,702- SD(LD)	33,648	34,355.99	21,676	27,825.62	SD(LD)	5%	35.58	4.03E-12
Exam5	132,953- LD	7,955	8,517.41	5,224	6,153.29	SD(LE)	5%	34.33	8.07E-27
Exam6	44,160-SD(LE)	29,515	30,703.08	26,165	27,519.42	SD(LWD)	5%	11.35	1.84E-27
Exam7	53,405- SD(LE)	12,568	13,745.50	7,201	7,453.69	SD(LWD)	10%	42.70	1.11E-30
Exam8	92,767 –SD(LE)	13,029	15,487.32	10,201	13,980.27	SD(LWD)	10%	21.71	0.005634

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

From Table 5, it is clear that PGH-HC is able to produce better results than TGH-HC approach for all instances. We obtain maximum improvement of 10.47% for tre-s-92 datasets followed by uta-s-92 and car-f-92 where improvement is around 10%. For the datasets car-s-91, kfu-s-93, rye-s-93, ute-s-92, and lse-f-91, we obtain at least 8% improvement (column I). It is also observed that combining SD with other heuristics produce better results (column G). Additionally, from the experiment, it is shown that for all instances 10% EAV produces best results. That is, smaller EAV is able to produce better results. It is also observed that *p*-values for all datasets are smaller than the level of significance of 0.05 .This means that there is a strong support for the hypothesis H_1, and we can say there is a significant evidence of better performance of the PGH-HC over TGH-HC for solving Toronto benchmark datasets.

6.2 Experimental Results with ITC2007 Datasets

Table 6(a)-(b) illustrate the characteristics of penalty values of ITC2007 datasets when two different termination criteria and two EAV are employed. Average costs as well as best costs for thirty individual runs using both TGH-HC and PGH-HC are highlighted. It is observed that in both TGH-HC and PGH-HC, with the same EAV, longer time duration is able to produce better penalty costs. Moreover, the increase of the EAV value within the same time duration leads to increase the penalty values for PGH-HC (i.e. bad quality solutions), but stable for TGH-HC because of its non-correlation with EAV. In general, PGH-HC performs better than TGH-HC for all datasets. However, their discrepancy is more apparent when lower EAV (i.e. 5%) and longer time duration (i.e. 3600 seconds) are employed.

A comparison between TGH-HC and PGH-HC for solving ITC2007 benchmark datasets is highlighted in Table 7. Best results are taken from Table 6(a)-(b) (results in bold font). For each instance, we have presented the best penalty cost and the average penalty cost. It is observed from the results that PGH-HC produces better result than TGH-HC for each instance of ITC2007 exam track. For Exam2, we have obtained improvement of more than 50% when comparing between PHG-HC and TGH-HC. For all other datasets except for Exam6 and Exam8 with an improvement percentage of 11.35 and 21.71 respectively, at least around 35% improvement is obtained. We notice that six out of eight datasets (except Exam7 and Exam8) produce good results with EAV of 5% (column H). This shows that, whenever possible, it would be betst to use small EAV. Finally, it is also observed from the *t*-test that *p*-values for all eight instances are smaller than the level of significance of 0.05, which indicate the validity of hypothesis H_1. Therefore, there is statistically significant evidence to support that PGH-HC approach produces better results than TGH-HC.

6.3 Comparison with the state-of-the-art approaches

Table 8 shows the comparison of the best result of the proposed PGH-HC approach with the state-of-the-art approaches for solving Toronto datasets. We have ranked the approaches according to their penalty costs in solving each instance and presented in the first bracket next to the corresponding penalty value. Average rank for each approach for solving overall Toronto datasets is also computed. Additionally, the best results are highlighted with bold font. It is observed that PGH-HC approach is able to solve the twelve instances of Toronto datasets. It is noticeable that approaches of Burke et al. [22], Caramia et al. [25], and Turabieh et al. [24] have better ranking and produce many good results. Although our approach does not produce any best result for any of the instances, the results are still comparable to the approaches in Table 8. In our approach, for sta-f-83 dataset, we obtain second best result. Moreover, our approach is ranked 3rd for car-f-92 and rye-s-93 datasets; fourth best result is obtained for three datasets and for rest of the instances decent rank is observed compared to other approaches. In average, the rank of the approach is fourth.

Table 9 shows the comparison of our approach with the five winners of the competition on ITC2007 datasets where termination criterion was set at 600 seconds. In general, our proposed PGH-HC approach has the ability to produce quality solution for all eight exams and for one instance (Exam6), where the best result is obtained. Six out of eight datasets are able to produce better result than Pillay [28]. Although De Smet[29] produced better result for five instances compared to our result, they, however, were unable to produce any result for three of the instances where we manage to produce a good result. Additionally, we also manage to produce better result in three instances compared to Atsuta et al. [30].

Finally, Table 10 compares our proposed approach with other reported results in the literature in solving ITC2007 exam datasets. Furthermore, the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ranking of each approach and the average results are presented. Here it is observed that in general McCollum et al. [31] and Abdullah et al. [32] produce the best results for most of the instances. This may be due to their longer termination criteria (more than 600 seconds). Although we tried to run longer (3600 seconds), our approach did not produce any best result; they are, however, comparable to the other approaches. We obtain the fifth best result for Exam6 and Exam8, whereas the seventh best for other instances except Exam5. If we compute average ranking, our approach is in the seventh position.

6.4 Discussions

Most of the approaches presented in the literature highlight population based meta-heuristics and hyper heuristics to solve the exam timetabling problem. In those presented methods, their approaches are complex and most often numerous parameter settings have to be tuned for producing good quality results. In our proposed approach, PGH-HC involves less and straight forward parameter tuning. Our only parameter is the EAV for the partial exam selection process and user can easily determine this value. From our experiment, usually smaller EAV value produced better result. Having smaller EAV value encourages the proposed approach to improve the partial solution further than having a larger EAV value. Although sometimes smaller EAV takes more computational time, the overall quality result can be obtained with reasonable time.

Table 8: Comparison of Our Approach with the State-of-the-art Results from the Literature for Toronto Datasets

Datasets	Carter et al. 1996[1]	Rahaman et al. 2014[23]	Turabieh et al. 2011[24]	Burke et al. 2012[22]	Caramia et al. 2008[25]	Pillay et al. 2009[26]	Sabar et al. 2012[27]	Our approach
car-s-91	7.10 (8)	5.12 (5)	4.80 (2)	**4.60**(1)	6.60 (7)	4.97 (3)	5.14 (6)	5.08 (4)
car-f-92	6.20 (8)	4.41(5)	4.10(2)	**3.90**(1)	6.00(7)	4.28(4)	4.70(6)	4.23(3)
ear-f-83	36.40(5)	36.91(6)	34.92(3)	32.80(2)	**29.30**(1)	35.86(4)	37.86(8)	37.06(7)
hec-s-92	10.80(4)	11.31(6)	10.73(3)	10.00(2)	**9.20**(1)	11.85(7)	11.90(8)	10.98(5)
kfu-s-93	14.00(3)	14.75(6)	**13.00**(1)	**13.00**(1)	13.80(2)	14.62(5)	15.30(7)	14.53(4)
lse-f-91	10.50(4)	11.41(7)	10.01(3)	10.0(2)	**9.60**(1)	11.14(6)	12.33(8)	11.12(5)
rye-s-93	7.30(2)	9.61(4)	9.65(5)	–(6)	**6.80**(1)	9.65(5)	10.71(6)	9.40(3)
sta-f-83	161.50(8)	157.52(3)	158.26(5)	**156.90**(1)	158.20(4)	158.33(6)	160.12(7)	157.23(2)
tre-s-92	9.60(8)	8.76(6)	**7.88**(1)	7.90(2)	9.40(7)	8.48(4)	8.32(3)	8.55(5)
uta-s-92	3.50(4)	3.54(5)	**3.20**(1)	**3.20**(1)	3.50(4)	3.40(2)	3.88(6)	3.43(3)
ute-s-92	25.80(3)	26.25(5)	26.11(4)	24.80(2)	**24.40**(1)	28.88(7)	32.67(8)	26.57(6)
yor-f-83	41.70(8)	39.67(5)	36.22(3)	**34.90**(1)	36.20(2)	40.74(7)	40.53(6)	38.83(4)
Average Rank (Round)	5.42(5)	5.25(5)	2.75(3)	1.83(2)	3.17(3)	5.00(5)	6.58(7)	4.25(4)

Table 9: Comparison with Five Winners on ITC2007 Competition Datasets

Datasets	Muller,2009 [33]	Gogs et al. 2008 [34]	Atsuta et al. 2008 [30]	De Smet 2008 [29]	Pillay 2008 [28]	Our approach (600 seconds)
Exam1	**4,370**	5,905	8,006	6,670	12,035	7,961
Exam2	**400**	1,008	3,470	623	3,074	1,716
Exam3	**10,049**	13,862	18,622	-	15,917	19,963
Exam4	**18,141**	18,674	22,559	-	23,582	22,211
Exam5	**2,988**	4,139	4,714	3,847	6,860	9,724
Exam6	26,950	27,640	29,155	27,815	32,250	**26,170**
Exam7	**4,213**	6,683	10,473	5,420	17,666	12,007
Exam8	**7,861**	10,521	14,317	-	16,184	12,482

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 10: Comparison of Our Approach with the State-of-the-art Results from the Literature for ITC2007 Datasets

Datasets	Muller 2009 [33]	Gogos et al. 2008 [34]	Atsuta et al 2008 [30]	De Smet 2008 [29]	Pillay,2008 [28]	McCollum et al. 2009 [31]	Rahman et al. 2014 [23]	Abdullah et al. 2012 [32]	Alzaqebah et al. 2015 [35]	Our approach (3600 seconds)
Exam1	4,370 (2)	5,905 (6)	8,006 (9)	6,670 (8)	12,035 (10)	4,633 (3)	5,231 (5)	**4,350** (1)	5,154 (4)	6,377 (7)
Exam2	400 (3)	1,008 (8)	3,470 (10)	623 (6)	3,074 (9)	390 (2)	433 (5)	**385** (1)	420 (4)	760 (7)
Exam3	10,049 (4)	13,862 (6)	18,622 (9)	- (10)	15,917 (8)	9,830 (2)	**9,265** (1)	9,951 (3)	10,182 (5)	15,092 (7)
Exam4	18,141 (5)	18,674 (6)	22,559 (8)	- (10)	23,582 (9)	17,251 (2)	17,787 (3)	18,000 (4)	**15,716** (1)	21,676 (7)
Exam5	**2,988** (1)	4,139 (7)	4,714 (8)	3,847 (6)	6,860 (10)	3,022 (2)	3,083 (4)	3,040 (3)	3,350 (5)	5,224 (9)
Exam6	26,950 (6)	27,640 (7)	29,155 (9)	27,815 (8)	32,250 (10)	**25,995** (1)	26,060 (3)	26,010 (2)	26,160 (4)	26,165 (5)
Exam7	4,213 (2)	6,683 (6)	10,473 (8)	5,420 (5)	17,666 (10)	**4,067** (1)	10,712 (9)	4,250 (3)	4,271 (4)	7,201 (7)
Exam8	7,861 (3)	10,521 (6)	14,317 (8)	- (10)	16,184 (9)	7,519 (2)	12,713 (7)	**7,450** (1)	7,922 (4)	10,201 (5)
Average Rank (Round)	3.25(3)	6.50(7)	8.63(9)	7.88(8)	9.38(9)	1.88(2)	4.63(5)	2.25(2)	3.88(4)	6.75(7)

7. CONCLUSIONS

In this research, our primary aim has been to develop a straight-forward approach to solve the examination timetabling problem. Hence, we have proposed combination of graph heuristic with hill climbing approach (PGH-HC) to address the problem. This approach is different as we incorporate partial construction and improvement approaches which attempt to schedule subsets of exams and improve these partially scheduled exams. Two benchmark datasets, which are Toronto un-capacitated and ITC2007 (exam track) capacitated datasets, have been incorporated to evaluate the proposed approach. Experimental results indicate that the proposed PGH-HC outperforms the TGH-HC for all instances on both datasets. It is also observed that PGH-HC is able to produce competitive results when compared with the state-of-the-art approaches reported in the scientific literature. Furthermore, different EAV and stopping criteria have been analysed and it reveal that smaller EAV value and longer improvement cycle (i.e. iterations, or time durations) is able to produce better result.

For future works, focus will be given to the further improvement of the proposed algorithm by including adaptive improvement cycle. We are optimistic that this will produce better results. We are also motivated to apply other meta-heuristic approaches, such as great deluge algorithm and simulated annealing to replace the hill climbing technique.

ACKNOWLEDGEMENTS:

This research is supported by Ministry of Posts, Telecommunication and Information Technology, Government of People's Republic of Bangladesh (Memo no: 56.00.0000.28.33.042.15-509) and the Graduate Research Scheme of University Malaysia Pahang (GRS 1403136).

REFERENCES:

[1] M. W. Carter, G. Laporte, and S. Y. Lee, "Examination timetabling: Algorithmic strategies and applications," *Journal of the Operational Research Society,* Vol. 47, No. 3, 1996, pp. 373-383.

[2] 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, No. 1, 2009, pp. 55-89.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[3] B. Hussin, A. S. H. Basari, A. S. Shibghatullah, S. A. Asmai, and N. S. Othman, "Exam timetabling using graph colouring approach," *IEEE Conference on Open Systems(ICOS)*, Lankawi(Malysia), September 25-28, 2011, pp. 133-138.

[4] L. Di Gaspero and A. Schaerf, "Tabu Search Techniques for Examination Timetabling," *Practice and Theory of Automated Timetabling III*. vol. 2079, Springer Berlin Heidelberg, 2001, pp. 104-117.

[5] J. M. Thompson and K. A. Dowsland, "A robust simulated annealing based examination timetabling system," *Comput. Oper. Res.,* vol. 25, No. 7,1998, pp. 637-648,

[6] M. M. Kahar and G. Kendall, "A great deluge algorithm for a real-world examination timetabling problem," *Journal of the Operational Research Society,*Vol.67, No.3, 2013, pp 116-133.

[7] E. K. Burke and Y. Bykov, "The late acceptance hill-climbing heuristic," *Technical report CSM-192 Computing Science and Mathematics*, University of Stirling, UK, 2012.

[8] E. K. Burke and J. P. Newall, "A multistage evolutionary algorithm for the timetable problem," *IEEE Transactions on Evolutionary Computation, ,* vol. 3, No.1,1999, pp. 63-74.

[9] M. Alzaqebah and S. Abdullah, "Artificial bee colony search algorithm for examination timetabling problems," *International Journal of Physical Sciences,* vol. 6, No. 17, 2011, pp. 4264-4272.

[10] S. Innet, "A noval approach of genetic algorithm for solving examination timetabling problems: A case study of Thai Universities," *13th IEEE International Symposium onCommunications and Information Technologies (ISCIT), Surat Thai (Thailand), September 4-6* , 2013, pp. 233-237.

[11] P. Boizumault, Y. Delon, and L. Peridy, "Constraint logic programming for examination timetabling," *The Journal of Logic Programming,* vol. 26, No. 2, 1996, pp. 217-233.

[12] E. K. Burke, S. Petrovic, and R. Qu, "Case-based heuristic selection for timetabling problems," *Journal of Scheduling,* Vol. 9, No.2, 2006, pp. 115-132.

[13] 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, No.4, 2009, pp. 981-1001.

[14] M. N. M. Kahar and G. Kendall, "A great deluge algorithm for a real-world examination timetabling problem," *Journal of the Operational Research Society,* Vol 67, No.3, 2013, pp.166-133.

[15] 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, No. 1, 2012, pp. 203-221.

[16] M. Ayob, 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* , 2006, p. 336-344.

[17] E. Burke, R. Qu, and A. Soghier, "Adaptive selection of heuristics for improving exam timetables," *Annals of Operations Research,*Vol. 218, No.1, 2012, pp. 1-17.

[18] N. R. Sabar, M. Ayob, G. Kendall, and Q. Rong, "Grammatical Evolution Hyper-Heuristic for Combinatorial Optimization Problems," *IEEE Transactions on Evolutionary Computation, ,* vol. 17, No.6, 2013, pp. 840-861.

[19] S. Ei Shwe, "Reinforcement learning with EGD based hyper heuristic system for exam timetabling problem,", *IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS)*, Beijing (China), 2011, pp. 462-466.

[20] B. McCollum, P. McMullan, E. K. Burke, A. J. Parkes, and R. Qu, "The second international timetabling competition: Examination timetabling track," *University of Nottingham, Queen's University, Notingham, Belfast, Technical Report: QUB/IEEE/Tech/ITC2007/Exam/v4. 0/17,* 2007.

[21] B. McCollum, A. Schaerf, B. Paechter, P. McMullan, R. Lewis, A. J. Parkes, *et al.*, "Setting the research agenda in automated timetabling: The second international timetabling competition," *INFORMS Journal on Computing,* vol. 22, No.1, 2010, pp. 120-130.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[22] E. Burke, N. Pham, R. Qu, and J. Yellen, "Linear combinations of heuristics for examination timetabling," *Annals of Operations Research,* vol. 194, No.1, 2012, pp. 89-109.

[23] 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, No. 2, 2014, pp. 287-297.

[24] 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.

[25] M. Caramia, P. Dell'Olmo, and G. F. Italiano, "Novel local-search-based approaches to university examination timetabling," *INFORMS Journal on Computing,* vol. 20, No. 1, 2008, pp. 86-99.

[26] 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, No. 2, 2009, pp. 482-491.

[27] N. Sabar, M. Ayob, R. Qu, and G. Kendall, "A graph coloring constructive hyper-heuristic for examination timetabling problems," *Applied Intelligence,* vol. 37, No. 1, 2012, pp. 1-11.

[28] N. Pillay, "A developmental approach to the examination timetabling problem," *Practice and theory of automated timetabling (PATAT 2008), 2008,* pp. 19-22.

[29] G. De Smet, "ITC2007—examination track," in *practice and theory of automated timetabling (PATAT 2008)*, Montreal, 2008.

[30] M. Atsuta, K. Nonobe, and T. Ibaraki, "ITC-2007 Track2: an approach using general CSP solver," 2008.

[31] 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,* August 10-12, 2009,pp. 424-434.

[32] S. Abdullah and H. Turabieh, "On the use of multi neighbourhood structures within a Tabu-based memetic approach to university timetabling problems," *Information Sciences,* vol. 191, No. 1, 2012, pp. 146-168.

[33] T. Müller, "ITC2007 solver description: a hybrid approach," *Annals of Operations Research,* vol. 172, No. 1, 2009, pp. 429-446.

[34] C. Gogos, P. Alefragis, and P. Housos, "Amulti-staged algorithmic process for the solution of the examination timetabling problem " *in Practice and Theory of Automated Timetabling (PATAT 2008),*Montreal, 2008, pp. 19-22.

[35] M. Alzaqebah and S. Abdullah, "Hybrid bee colony optimization for examination timetabling problems," *Computers & Operations Research,* vol. 54, No. 1, 2015, pp. 142-154.