Research Journal of Applied Sciences 8 (2): 131-138, 2013
ISSN: 1815-932X

Hybrid Approach: Tabu-Based Non-Linear Great Deluge for the Course Timetabling Problem

Rohazlin Md Yassin, Mohd Zakree Ahmad Nazri and Salwani Abdullah
Center for Artificial Intelligence Technology, Faculty of Information Science and Technology,
University Kebangsaan Malaysia, 43600 UKM, Bangi Selangor, Malaysia

Abstract: In this study, researchers present a hybridisation of Tabu-based technique and Non-Linear Great Deluge algorithm applied on course timetabling problems. This approach uses a short term memory in Tabu-based technique in order to improve the searching process. The Tabu list exercised is in the ranges of two to eight in length. The experiment results demonstrate that this hybrid approach is capable to produce competitive results when compared with others in the literature.

Key words: Hybrid metaheuristic, course timetabling, Tabu list, non-linear great delugem, memory

INTRODUCTION

Timetable is an organized list that usually provides information about a series of arranged events, particularly, the time at which it is planned that the particular events will take place. Timetabling is known to be a highly constrained combinatorial optimization problem (NP-hard) and has been studied in some details over the last few decades. Most of the optimization problems (NP-hard) are difficult to solve to optimality (Klein and Young 1999; Cormen *et al.*, 2002). Extensive efforts have been carried out in developing effective timetabling. It is the norm that the traditionally created timetable is based on trial and error basis and hence, the solution produced is not always guaranteed.

There are various types of timetabling which are not exhaustive only to educational timetabling such as employee timetabling, nurse timetabling, sports timetabling, communication timetabling and transportation timetabling. The timetabling problems basically consist of a set of resources, a set of activities, a set of dependencies between the activities and the time that is divided into time slots. Scheduling a timetabling is subject to constraints that are usually divided in two categories which are either hard or soft constraints. Hard constraints must be strictly satisfied with no violation allowed while in the case of soft constraints it is desirable but not essential to minimize violations. It is the soft constraints which effectively define how good a given feasible solution is so that different solutions can be compared and improved via an objective (fitness) function. However, the constraints and their importance differ significantly among countries and among institutions (Carter and Laporte, 1997).

University timetabling is probably one of the best studied timetabling problems in academia. The task in preparing the course timetabling is typically a real-world scheduling problem in every university. This problem must be solved by the university administration every year or even every term and it involves a large amount of human and material resources. Scheduling a number of lectures for each course within a given number of rooms and timeslots is known as the university course timetabling. Given the large number of events (lectures) to be scheduled and the diversity of constraints, the problems to be solved are very difficult.

In this research, the hard and soft constraints considered are as presented by Socha *et al.* (2002). The hard constraints are student and lecturers cannot be in two places at the same time, only one course is allowed to be in a timeslot in each classroom, the classroom capacity will be equal to or greater than the number of students attending the course at a particular timeslot and finally, the classroom assigned to the course should satisfy the features required by the course. Meanwhile, the soft constraints considered are students should not be scheduled a single course in a day, students should not have more than two consecutive courses on a day and students should not be scheduled a course in the last timeslot of the day.

Corresponding Author: Rohazlin Md Yassin, Center for Artificial Intelligence Technology, Faculty of Information Science and Technology, University Kebangsaan Malaysia, 43600 UKM, Bangi Selangor, Malaysia

In the past, a wide variety of approaches have been investigated, developed and tested for solving the various course timetabling problems including ant colony algorithm, simulated annealing, Tabu search, great deluge and mimetic algorithm.

HYBRID APPROACH: RESEARCH AND DEVELOPMENTS

A hybrid approach is one which includes two or more methods collectively. The advantage of combining two or more methods together can potentially helps to reduce the insufficiency of using only one method in segregation.

Evolutions towards hybridisation among the algorithms have proven to produce excellent results. Abdullah *et al.* (2007b) presented a hybrid approached combining a mutation operator with their earlier randomized iterative improvement procedure and has produced new best result. Previously, Abdullah *et al.* (2005) has proposed several versions of Variable Neighbourhood Search (VNS), one VNS-basic, VNS-Monte-Carlo acceptance criterion and hybridise VNS with Tabu Search (VNS-Tabu). The version of hybridising VNS-Tabu produces the best result for three of the small instances tested.

Among the many approaches, meta-heuristics are the most used techniques and the current trend indicates that multi-stage or hybrid meta-heuristics are the most successful approaches that can be attempted. Meta-heuristic approaches have attracted the most attention in solving the university timetabling problem. This is due to the ability of these approaches in generating good solutions. One of the latest known meta-heuristic approaches is the modified version of the Great Deluge algorithm called the Non-Linear Great Deluge (NLGD) algorithm introduced by Landa-Silva and Obit (2008).

From the initial study, it is understood that the Tabu-based technique has an advantageous characteristic by which it could prevent the local search from wondering in the local optima. This is what that initially draws the interest to investigate whether the Tabu-based technique can further assist the NLGD algorithm from getting stuck in the local optima while able to improve the timetabling solution.

PROBLEM DESCRIPTION

The problem description employed as part of this research is adapted from the descriptions given by Rossi-Doria *et al.* (2002) and Socha *et al.* (2002). The problem can be stated as follows:

- A set of N courses labeled as $\{e_1, e_2, e_3, \ldots, e_N\}$
- A set of T number of timeslots which is equal to 45 (5 days with 9 timeslot/day)
- A set of M number of students
- A set of R number of rooms
- A set of F number of features

The cost function adopted from Lewis (2008) is as defined in Eq. 1 and is used to calculate the quality of the timetabling solution:

$$f(T) = \sum_{i=1}^{k} w_i v_i(T) \tag{1}$$

Where:

- T = The given timetable
- k = The type of constraints
- w_i = The penalty weighted associated with each constraint i
- $v_i(T)$ = The number of constraints violations of type i in a timetable type T

A HYBRIDIZATION OF Tabu-BASED TECHNIQUE AND NON-LINEAR GREAT DELUGE ALGORITHM

Tabu-based technique: Tabu-based technique which was developed by Glover (1986) demonstrated its ability to solve large and difficult combinatorial optimization problems and has performed well especially in educational timetabling. It is specifically designed to avoid the trap of local optimality by utilizing a short-term memory in order to prevent the return to inverse moves (cycling) while utilizing the long term memory in order to diversify and intensify the search space.

Tabu-based is a meta-heuristic algorithm that can be used to solve combinatorial optimization problems and is a global heuristic technique which tries to avoid falling into local optima by creating a special list called Tabu (Reeves, 1993). In several cases, Tabu-based is described to provide solutions which are very close to optimality and are among the most effective if not the best to tackle the difficult problems at hand (Gendreau, 2002). Tabu-based is an iterative search procedure that starts from an initial feasible solution which then progressively improves the solution by applying a series of moves. Any solution which has recently been selected is put into a Tabu list so that it becomes ‘taboo’ for a short period of time, depending on the length of the Tabu list. These minimize the chance of cycling in the same solution and therefore create more chances of improvement by moving into un-explored areas of the search space. The Tabu-based technique’s search space is simply the space

Table 1: The neighbourhood structures and corresponding Tabu list

Neighbourhood structure	Description	Tabu list
M1	Select one course at random and assign to a feasible pair timeslot-room that is chosen at random	T_List1
M2	Identifies a course that violates soft constraints and move it to another randomly selected pair timeslot-room and ensuring feasibility is maintained	T_List2
M3	Select two courses at random and swap their timeslots and rooms while maintaining the feasibility of the solution	T_List3

of all possible solutions that can be visited during the search whereas the Tabu list records the recent history of the search (Table 1).

Non-Linear Great Deluge algorithm (NLGD): The NLGD was introduced by Landa-Silva and Obit (2008) which is an extension of the Great Deluge (GD) algorithm which was introduced by Dueck (1993). GD algorithm needs just two parameters: the amount of computational time that the user wishes to spend and an estimate of the quality of solution that a user requires. In GD algorithm when the new candidate solution S^* is worse than the current solution S then S^* replaces S depending on the current water level where the water level is identified as B. The decay rate is the speed at which B decreases and is determined by a linear function in the original great deluge algorithm. The water level decay rate is pre-determined and fixed whereas for NLGD, the decay rate of the water level is controlled by an exponential function. The NLGD exponential function for the water level is given by Eq. 2:

$$B = B\times\exp(-\delta(\text{rnd}[\min, \max]))+\beta \tag{2}$$

The algorithm basically concerns with the decay rate of the water level in a non-linear manner. Equation 2 is used to control the speed and the shape of the water level decay rate. The NLGD acceptance criterion accepts the improving and non-improving low-level heuristics depending of the performance of the heuristic and the current water level B. Improving heuristics are always accepted while non-improving ones are accepted only if the detriment in quality is less than or equal to B. The initial water level is usually set to the quality of the initial solution and then decreased by a non-linear function. The parameter β function influences the shape of the decay rate and it represents the minimum expected penalty corresponding to the best solution. Whereas the role of the parameters min and max as in the expression (exp (-δ (rnd [min, max]))) is to control the speed of the decay rate and hence, the speed of the search process. It is expected that the higher the values of min and max, the faster the water level goes down and in consequences, the search quickly achieves improvement.

In this research, all the parameter settings related to NLGD algorithm are adopted as defined by Landa-Silva and Obit (2008) as shown in Algorithim 1.

Algorithm 1 (Pseudo-code for the NLGD algorithm):

```
Construct initial feasible solution S
Set best solution so far S_best←S
Set timeLimit according to problem size
Set initial water level B←f (S)
while elapsedTime ≤ timeLimit do
   Select move at random from M1, M2, M3
   Define the neighbourhood N(S) of S
   Select candidate solution S*∈N(S) at random
   if (f (S*)≤f (S) or f (S*)≤B ) then
   S←S* {accept new solution}
   S_best ←S {update best solution}
end if
range = B-f (S*)
if (range<1) then
   if (Large or Small Problem) then
      B = B+rand [Bmin,Bmax]
    else
      if (f (S_best)<flow ) then
         B = B+rand [Bmin,Bmax]
      else
         B = B+2
      end if
    end if
else
   B = B×(exp-δ (rnd [min,max]))+β
 end if
end while
```

A HYBRID APPROACH

The hybrid approach between the Tabu-based technique and NLGD algorithm is divided into two parts, i.e., Part 1 Initialisation and Part 2 Tabu-based NLGD algorithm. In Part 1, the quality of the initial and best solutions are calculated based on the set duration of time. In Part 2, the Tabu-based NLGD algorithm is implemented.

The applied Tabu tenure is a fixed Tabu length of ranges from two to eight (Tabu tenure of length 2-8). Three individual Tabu lists is utilized representing the three neighbourhood structures where each neighbourhood is assigned to the individual Tabu list. An event that has improved the penalty cost of the timetable is then inserted in the Tabu list. The data stored in the Tabu list is the event number corresponding to the improved neighbourhood move. When the length of the Tabu list reaches the maximum length (Tabu tenure), the oldest event that resides in the Tabu list is released following a simple First In First Out (FIFO) Method. The events in the Tabu list are prohibited to be chosen for the next move for a certain number of iterations or until the event is removed from the Tabu list. This will give more opportunity to other events to be considered in performing the moves that may result in better quality solutions. This is also as an attempt to avoid from the cyclic moves. The Tabu tenure is assigned to the same length for every neighbourhood at any one time. Table 1 shows the neighbourhood structures and corresponding Tabu list representation of this research.

The neighbourhood move of T-List1 is a matrix of (Tabu tenure length×1) dimensional array, T-List2 is also matrix of (Tabu tenure length×1) dimensional array while T-List3 is a matrix of (Tabu tenure length×2) dimensional array. Let say, one event is selected at random. Then, at any successful iteration, the inserting and removing of the event from the Tabu list for neighbourhood move (M1 and M2) deals with only one item at a time. It is slightly different with neighbourhood move M3 since two events are selected simultaneously. In any successful iteration, two events are inserted in the Tabu list T-List3. The same goes with removing the items in the Tabu list where the two events will be removed. The pseudo code for the Tabu-based NLGD algorithm is presented in Algorithim 2.

Algorithm 2 (Pseudo-code for the Tabu-based NLGD algorithm):

```
Construct initial feasible solution S
Initialise S_best = S,
Initialise Beta = 0;
Initialise initial water level, B = f (S),
Initialise DecayRate = expδ (rnd [min,max]) + Beta;
Initialise T_List1, T_List2, T_List3
case small problem datasets
    TimeLimit = 3600; MaxRoom = 5; Bmin = 2; Bmax = 5;
    Sigma = 5*(10^-10); Min = 10000; Max = 20000;
case medium problem datasets
    TimeLimit = 7200; MaxRoom = 10; Bmin = 1; Bmax = 4;
    Sigma = 5*(10^-8); Min = 100000; Max = 300000;
case large problem datasets
    TimeLimit = 10800; MaxRoom = 10; Bmin = 1; Bmax = 3;
    Sigma = 5*(10^-9); Min = 100000; Max = 300000;
end-case
while ((elapsedTime≤timeLimit) and (Waterlevel>Beta))do
     Select move at random from M1, M2, M3
     Generate feasible solution, S
     if (f(S*)≤ f(S) or f(S*)≤B) then
         S = S*{accept new solution}
         S_best = S{update best solution}
    Remove first item from respective Tabu_list if T_Length
        > = Max T_List
  Insert event into the respective Tabu list, T_List1,
     T_List2   or T_List3
end if
range = B-f (S*)
if (range<1) then
   if (Large or Small Problem) then
       B = B+rand [Bmin,Bmax]
 else
   if (f(S_best)<flow) then
       B = B+rand [Bmin, Bmax]
     else
        B = B + 2
     end if
 end if
else
   B = B × (exp-δ (rnd [min, max])) + β
 end if
end while
```

SIMULATION RESULTS

The Tabu-based NLGD algorithm was programmed using MATrix LABoratory, MATLAB tools on a personal computer with Intel® Core 2 Duo CPU at the rate 3.00 GHz with 2 GB of RAM using Windows VistaTM Home Basic. Table 2 shows the results for the hybrid algorithm compared to the NLGD algorithm after some preliminary experiments.

In order to ensure a fair comparison, the NLGD algorithm is re-developed using the pseudo-code of Algorithim 1 by Landa-Silva and Obit (2008) with the same duration of time and CPU processing capability as applied to the hybrid approach.

For each type of problem dataset a fixed computation time (sec) was used as the termination criteria condition: 3600 for small problems, 7200 for medium problems and 10800 for the large problems. This fixed computation time is only for the improvement phase, i.e., the Non-Linear Great Deluge starting from an initial feasible solution. For each problem instance we executed the NLGD and hybrid approach algorithms for 10 times. Table 2 shows the result obtained from the NLGD alongside other results by the hybrid approach algorithms.

The best results are shown in bold for each dataset. The main goal of this comparison is to assess whether NLGD algorithm performs better with the hybridization of the Tabu-based techniques. The result indicates that the NLGD algorithm is able to produce better timetabling solution with the hybridization. The approach is best suited at Tabu length 6 since nine out of eleven best ten runs produce lowest penalty cost. Figure 1 shows the convergence of the hybrid approach. The graphs in Fig. 1 show the behavior of the Tabu-based NLGD when tested on small2, medium3 and large datasets. Tabu-based NLGD is able to obtain better convergence in comparisons with NLGD algorithm that contributes to better solution quality.

Table 2: The results of the Tabu-based NLGD with different Tabu Length (TL) in comparisons with the NLGD algorithm

		Tabu-based NLGD						
Dataset	NLGD	TL = 2	TL = 3	TL = 4	TL = 5	TL = 6	TL = 7	TL = 8
s1	2	1	0	0	1	0	1	1
s2	1	0	0	0	0	0	0	0
s3	2	2	1	1	1	0	1	1
s4	2	0	1	1	0	1	1	1
s5	0	0	0	0	0	0	0	0
m1	122	91	88	95	89	79	93	90
m2	128	95	89	94	88	83	89	94
m3	167	103	107	103	104	99	101	102
m4	130	93	91	88	90	97	97	91
m5	138	99	87	95	84	74	90	89
l	726	491	476	491	467	458	528	565

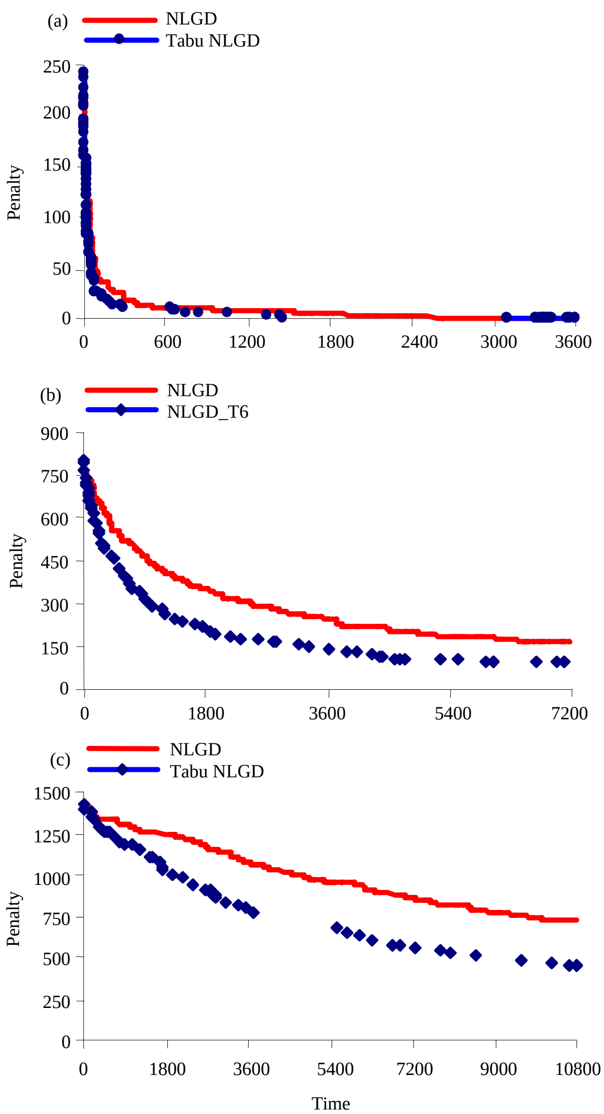

Fig. 1: The convergence of the Tabu-based NLGD algorithm; a) S^2; b) m^3; c) 1

Furthermore, researchers are particularly interested to compare the results with the other results in the literature as shown in Table 3, i.e.:

A1 = Hybrid harmony search algorithm by Al-Betar and Khader (2009)
A2 = Adapted harmony search algorithm by Al-Betar *et al.* (2008)
A3 = Modified harmony search algorithm by Al-Betar *et al.* (2012)
A4 = Harmony search algorithm with multi pitch adjusting rate by Al-Betar *et al.* (2010)
A5 = Random restart local search by Socha *et al.* (2002)
A6 = MAX-MIN ant system by Socha *et al.* (2002)
A7 = Tabu-search hyper-heuristic by Burke *et al.* (2003)
A8 = Variable neighbourhood search by Abdullah *et al.* (2005)
A9 = Fuzzy multiple heuristic ordering by Asmuni *et al.* (2005)
A10 = Distributed choice function hyper-heuristic by Gaw *et al.* (2005)
A11 = Graph-based hyper-heuristic by Burke *et al.* (2007)
A12 = Randomized iterative improvement by Abdullah *et al.* (2007b)
A13 = Great deluge by Landa-Silva and Obit (2008)
A14 = Non-linear great deluge by Landa-Silva and Obit (2008)
A15 = Non-linear great deluge hyper-heuristic with static memory by Obit *et al.* (2009)
A16 = Non-linear great deluge hyper-heuristic with dynamic memory by Obit *et al.* (2009)
A17 = Great deluge and Tabu search by Abdullah *et al.* (2009)
A18 = Particle collision algorithm by Abuhamdah and Ayob (2009)
A19 = Late acceptance randomized descent by Abuhamdah (2010)
A20 = Extended great deluge by McMullan (2007)
A21 = Hybrid evolutionary approach by Abdullah *et al.* (2007a)
A22 = Mimetic algorithm by Jang *et al.* (2008)
A23 = Evolutionary non-linear great deluge by Landa-Silva and Obit (2009)
A24 = Electromagnetism mechanism great deluge by Turabieh *et al.* (2009)
A25 = Guided genetic algorithm by Jat and Yang (2009)
A26 = Ant colony system with simulated annealing by Ayob and Jaradat (2009)
A27 = Ant colony system with Tabu search by Ayob and Jaradat (2009)
A28 = Genetic algorithms with guided and local search by Yang and Jat (2011)

From Table 3, researchers can see that the algorithm is comparably better when compared against other methods. Based on the total penalty, the approach is ranked second (with the total penalty equal to 891) after the approach by Al-Betar and Khader (2009) at the first rank (with the total penalty = 843). The presented results show that the approach is able to work well across different problems size which represents different complexity of the problem. It can be observed that some of the approaches in comparison only work well on the small datasets whilst they show a poor performance for

Table 3: Results comparisons between Tabu-based NLGD with state of the art approaches

Data sets	s1	s2	s3	s4	s5	m1	m2	m3	m4	m5	l	Total penalty	Rank
Tabu-based NLGD	0	0	0	1	0	79	83.0	99	97.0	74.0	458.0	891.0	2
A1	0	0	0	0	0	99	73.0	130	105.0	53.0	383.0	843.0	1
A2	3	2	4	3	0	223	216.0	272	202.0	177.0	-	-	-
A3	0	0	0	0	0	168	160.0	176	144.0	71.0	417.0	1136.0	5
A4	0	0	0	0	0	124	117.0	148	132.0	67.0	424.0	1012.0	4
A5	8	11	8	7	5	199	202.5	-	177.5	-	-	-	-
A6	1	3	1	1	0	195	184.0	284	164.5	219.5	851.5	1904.5	19
A7	1	2	0	1	0	146	173.0	267	169.0	303.0	1166.0	2228.0	21
A8	0	0	0	0	0	317	313.0	375	247.0	292.0	-	-	-
A9	10	9	7	17	7	243	325.0	249	285.0	132.0	1138.0	2422.0	22
A10	1	3	1	1	0	182	164.0	250	168.0	222.0	-	-	-
A11	6	7	3	3	4	372	419.0	359	348.0	171.0	1068.0	2760.0	23
A12	0	0	0	0	0	242	161.0	265	181.0	151.0	-	-	-
A13	17	15	24	21	5	201	190.0	229	154.0	222.0	1066.0	2144.0	20
A14	3	4	6	6	0	140	130.0	189	112.0	141.0	876.0	1607.0	16
A15	0	0	0	0	0	71	82.0	137	55.0	106.0	777.0	1228.0	9
A16	0	0	0	0	0	88	88.0	112	84.0	103.0	915.0	1390.0	11
A17	0	0	0	0	0	78	92.0	135	75.0	68.0	556.0	1004.0	3
A18	1	1	1	1	0	136	138.0	165	143.0	135.0	789.0	1510.0	14
A19	0	0	0	0	0	149	132.0	200	138.0	173.0	855.0	1647.0	17
A20	0	0	0	0	0	80	105.0	139	88.0	88.0	730.0	1230.0	10
A21	0	0	0	0	0	221	147.0	246	165.0	130.0	529.0	1438.0	12
A22	0	0	0	0	0	227	180.0	235	142.0	200.0	-	-	-
A23	0	1	0	0	0	126	123.0	185	116.0	129.0	821.0	1501.0	13
A24	0	0	0	0	0	96	96.0	135	79.0	87.0	683.0	1175.0	6
A25	0	0	0	0	0	240	160.0	242	158.0	124.0	801.0	1725.0	18
A26	0	0	0	0	0	117	121.0	158	124.0	134.0	645.0	1199.0	8
A27	0	0	0	0	0	150	179.0	183	140.0	152.0	750.0	1554.0	15
A28	0	0	0	0	0	139	92.0	122	98.0	116.0	615.0	1182.0	7

the bigger size of problems (such as A19, A25 and A27). Researchers believed that the combination of the Tabu list with the non-linear great deluge helps the algorithm to escape from local optima and jump to different search area. This is due to the Tabu list that keeps the event/course that has reduced the penalty cost, allows other event/course to be selected and explored in the next iterations which later contribute to better quality of the solutions. In general, researchers can conclude that the proposed approach is competent in obtaining competitive results in comparisons with other approaches in the literature.

CONCLUSION

This study presents a hybridization approach that combined Tabu-based technique and NLGD algorithm. To the knowledge, this is the first such algorithm aimed at this problem domain. In order to test the performance of the approach, experiments are carried out based on the course timetabling problems and compare with a set of state of the art methods from the literature. Even so, there are still many areas can be improved. Further exploration such as implementing a dynamic Tabu-based technique to the NLGD algorithm rather than static Tabu as currently proposed in this approach is strongly recommended. The dynamic Tabu tenure method by varying the Tabu list during the running of the algorithm approach is worth investigating in the future research. Researchers also believed that this proposed approach should be tested on other combinatorial optimisation problems (not only restricted to course or exam timetabling) to better measure the performance of this algorithm.

This approach is simple yet effective and able to produce comparable (with one best) results in comparisons with other approaches studied in the literature. As an overall remark, the experimental evidence shows that by hybridising the Tabu-based technique to NLGD algorithm, research have been able to produce a hybrid approach that is still quite simple but much more effective in generating solutions for a well-known set of difficult course timetabling problem instances.

REFERENCES

Abdullah, S., E.K. Burke and B. McCollum, 2007a. Using a randomized iterative improvement algorithm with composite neighborhood structures. Proceeding of the 6th International Conference on Meta-Heuristic, September 25-28, 2007, Singapore, pp: 153-169.

Abdullah, S., E.K. Burke and B. McCollum, 2007b. A hybrid evolutionary approach to the university course timetabling problem. Proceeding of the IEEE Congres on Evolutionary Computation, September 25-28, 2007, Singapore, pp: 1764-1768.

Abdullah, S., E.K. Burke and B. McColum, 2005. An investigation of variable neighbourhood search for university course timetabling. Proceedings of the 2nd Multidisciplinary International Conference on Scheduling: Theory and Applications, July 18-21, 2005, New York, USA., pp: 413-427.

Abdullah, S., K. Shaker, B. McCollum and P. McMullan, 2009. Construction of course timetables based on great deluge and Tabu search. Proceedings of the 8th International Conference on Metaheuristic, July 13-16, 2009, Hamburg, Germany.

Abuhamdah, A. and M. Ayob, 2009. Experimental result of particle collision algorithm for solving course timetabling problems. Int. J. Comput. Sci. Network Secur., 9: 134-142.

Abuhamdah, A., 2010. Experimental result of late acceptance randomized descent algorithm for solving course timetabling problems. Int. J. Comput. Sci. Network Secur., 10: 192-200.

Al-Betar, A.M. and A.T. Khader, 2009. A hybrid harmony search for university course timetabling. Proceedings of the Multidisciplinary International Conference on Scheduling: Theory and Applications, August 10-12, 2009, Dublin, Ireland.

Al-Betar, A.M., A.T. Khader and T.A. Gani, 2008. A harmony search algorithm for university course timetabling. Proceedings of the 7th International Conference on the Practice and Theory of Automated Timetabling, August 18-22, 2008, Montreal, Canada.

Al-Betar, M.A., A.T. Khader and M. Zaman, 2012. University course timetabling using a hybrid harmony search metaheuristic algorithm. IEEE Trans. Syst. Man Cybernet. Part C: Appli. Rev., 42: 664-681.

Al-Betar, M.Z., A.T. Khader and I.Y. Liao, 2010. A harmony search with multi-pitch adjusting rate for the university course timetabling. Recent Adv. Harmony Search Algorithm Stud. Comput. Intell., 2010: 147-161.

Asmuni, H., E.K. Burke and J.M. Garibaldi, 2005. Fuzzy multiple heuristicordering for course timetabling. Proceeding of the 5th United Kingdom Workshop on Computational Intelligence, September 5-7, 2005, London, pp: 302-309.

Ayob, M. and G. Jaradat, 2009. Hybrid ant colony systems for course timetabling problems. Proceedings of the 2nd IEEE Conference on Data Mining and Optimization, October 27-28, 2009, Kajand, pp: 120-126.

Burke, E.K., B. McCollum, A. Meisels, S. Petrovic and R. Qu, 2007. A graph-based hyper-heuristic for educational timetabling problems. Eur. J. Oper. Res., 176: 177-192.

Burke, E.K., G. Kendall and E. Soubeiga, 2003. A Tabu-search hyper-heuristic for timetabling and rostering. J. Heuristics, 9: 451-470.

Carter, M. and G. Laporte, 1997. Recent developments in practical course timetabling. Proceedings of the 2nd International Conference on Practice and Theory of Automated Timetabling II, August 20-22, 1997, Toronto, Canada, pp: 3-19.

Cormen, T.H., C.E. Leiserson, R.L. Rivest and C. Stein, 2002. Introduction to Algorithms. MIT Press, Cambridge, UK.

Dueck, G., 1993. New optimization heuristics: The great deluge algorithm and the record-to-record travel. J. Comput. Phys., 104: 86-92.

Gaw, A., P. Rattadilok and R.S. Kwan, 2005. Distributed Choice Function Hyperheuristics for Timetabling and Scheduling. In: Practice and Theory of Automated Timetabling V, Burke, E.K. and M. Trick (Eds.). Springer-Verlag, Berlin, Germany, pp: 51-67.

Gendreau, M., 2002. An Introduction to Tabu Search. In: Handbook of Metaheuristics, Glover, F. and G. Kochenberger (Eds.). Kluwer, Norwell, MA., USA., pp: 37-54.

Glover, F., 1986. Future paths for integer programming and links to artificial intelligence. Comput. Operat. Res., 13: 533-549.

Jang, W.S., H. Kang and B.H. Lee, 2008. Hybrid simplex-harmony search method for optimization problems. Proceedings of the IEEE Congress on Evolutionary Computation, June 1-6, 2008, Hong Kong, pp: 4157-4164.

Jat, S.N. and S. Yang, 2009. A guided search genetic algorithm for the university course timetabling problem. Proceedings of the 4th Multidisciplinary International Conference on Scheduling: Theory and Applications, August 10-12, 2009, Dublin, Ireland, pp: 180-191.

Klein, P. and N. Young, 1999. On the number of iterations for Dantzig-Wolfe decomposition. Comput. Oper. Res., 26: 1195-1216.

Landa-Silva, D. and J.H. Obit, 2008. Great deluge with nonlinear decay rate for solving course timetabling problems. Proceedings of the 4th International IEEE Conference on Intelligent Systems, September 6-8, 2008, Varna, Bulgaria, pp: 8.11-8.18.

Landa-Silva, D. and J.H. Obit, 2009. Evolutionary Non-Linear Great Deluge for University Course Timetabling. In: Hybrid Artificial Intelligence Systems, Corchado, E., X. Wu, E. Oja, E. Hristozov and T. Jedlovcnik (Eds.). Springer-Verlag, Berlin, Germany, pp: 269-276.

Lewis, R., 2008. A survey of metaheuristic-based techniques for University Timetabling problems. OR Spectrum, 30: 167-190.

McMullan, P., 2007. An extended implementation of the great deluge algorithm for course timetabling. Proceedings of the Part 1, 7th International Conference on Computer Science, May 27-30, 2007, Beijing, China, pp: 538-545.

Obit, J., D. Landa-Silva, D. Ouelhadj and M. Sevaux, 2009. Non-linear great deluge with learning mechanism for solving the course timetabling problem. Proceedings of the 8th International Conference on Metaheuristic, July 13-16, 2009, Hamburg, Germany, pp: 1-10.

Reeves, C.R., 1993. Improving the efficiency of Tabu search for machine sequencing problems. J. Oper. Res. Soc., 44: 375-382.

Rossi-Doria, O., C. Blum, J. Knowles, M. Samples, K. Socha and B. Paechter, 2002. A local search for the timetabling problem. Proceedings of the 4th International Conference on the Practice and Theory of Automated Timetabling, August 21-23, 2002, Gent, Belgium, pp: 124-127.

Socha, K., J. Knowles and M. Samples, 2002. A max-min ant system for the university course timetabling problem. Proceedings of the 3rd International Workshop on Ant Algorithms, Lecture Notes in Computer Science, September 12-14, 2002, Springer Verlag, Germany, pp: 1-13.

Turabieh, H., S. Abdullah and B. McCollum, 2009. Electromagnetism-like mechanism with force decay rate great deluge for the course timetabling problem. Rough Sets Knowledge Technol., 5589: 497-504.

Yang, S. and S.N. Jat, 2011. Genetic algorithms with guided and local search strategies for university course timetabling. IEEE Trans. Syst. Man Cybernetics PART C: Applicat. Rev., 41: 67-82.