Research Article

Fouad H. Awad, Ali Al-kubaisi, and Maha Mahmood*

Large-scale timetabling problems with adaptive tabu search

https://doi.org/10.1515/jisys-2022-0003
received September 26, 2021; accepted November 15, 2021

Abstract: Timetabling problems are specific types of scheduling problems that deal with assigning certain events to the timeslots. This assigning is subject to certain hard constraints that should be achieved to get a feasible timetable and soft constraints that must meet as many as possible during forming a feasible schedule. This paper introduces an adaptive tabu search. Eleven benchmark datasets of the year 2002 are applied to show the effectiveness of the introduced algorithm. These datasets consist of 5-small, 5-medium, and 1-large dataset. As compared to other methods from previous works, the proposed algorithm produces excellent timetables, in comparison with the algorithms as well as the current results, the mathematical results showed the high effectiveness of the suggested algorithm. It has a minor deficit on the medium or the small problem adaptive Tabu, and the tabu search relies on the tabu list and penalty cost when the change in the penalty cost is checked; if it is still unchanged for the period of iterations (1,000 iterations), the tabu list reduces automatically by (–2); furthermore, the tabu list remains constant.

Keywords: meta-heuristics, course timetabling, tabu search

1 Introduction

In reality, the timetabling problem is a challenging task in constructing a well-behaved solution [1]. The problem's size represents the main difficulty. This difficulty includes a sizeable volume of rooms, courses, teachers, and students that are correlated in several ways throughout conditions and objectives. Thus, a huge amount of variables and constraints should be taken into consideration during the procedure of each solution [2]. In the problem of course timetabling, a heuristic algorithm is applied to obtain a feasible starting solution. It begins by assigning a course to time slots. Scheduling the course is the first step in the strategy of starting. Since the course involves a sizeable number of students, it is very hard to reschedule. To place a course in the best appropriate time slots, it is recommended to construct a schedule based on a blank timetable [3]. This technique is known as least saturation degree. The academic institution faces the problem of scheduling appointments of the courses and examinations in every semester. This problem raises due to the increasing number of students, courses, exams, rooms, and the invigilator constraints. An automated timetabling system can produce functional and accurate schedules; the types of timetabling at educational institutions (schools and universities) are manual, semi-automated, and fully automated timetabling. A survey conducted by Iyad Abu Doush et al. (2008) in 56 universities in the UK discovered that only 58% (32 universities) uses the computer for preparing their examination timetable while 21% (11 universities) has a scheduling system [4]. Hard requirements must be firmly adhered to, and they take

*** Corresponding author: Maha Mahmood,** College of Computer Science & Information Technology, University of Anbar, Anbar, Iraq, e-mail: maha-mahmood@uoanbar.edu.iq
Fouad H. Awad: College of Computer Science & Information Technology, University of Anbar, Anbar, Iraq, e-mail: fouad.hammadi@uoanbar.edu.iq
Ali Al-kubaisi: Computer Center University of Anbar, Anbar, Iraq, e-mail: alijamal530@gmail.com

precedence over soft constraints. If a timetable meets all hard restrictions, it is said to be feasible. Event clash limitations are likely to be prevalent hard constraints in educational scheduling issues. These constraints typically occur when a single resource is requested by two events at the same time, which is impossible to do. Almost every educational institution has this constraint [5].

This paper presents the problem of university course timetabling examined using an adaptive tabu search algorithm to enhance the initial feasible solution. Eleven benchmark datasets presented by Socha et al. are employed for testing this approach. The findings appear that our method can produce excellent solutions in comparison with other methods in the literature.

This paper contributes to developing a tabu search algorithm, an adaptive tabu list to be compatible resulting in a solution for the scheduling a timetable problem.

The structure of this paper is logically organized where the next section portrays the related works. A brief explanation of the course-timetabling problem is given in Section 3.

The meta-heuristic algorithm and its application to the course-timetabling problem are described in Section 4. Section 5 presents experimental results and a comparison between techniques from the related work, and Section 6 reports brief concluding comments.

2 Related works

Solving problems of course timetabling and specifically enrolment-based have been proposed in the last three decades. Four categories of automatic timetabling procedures can be extracted from wide-ranging literature reviews. The first category deals the timetabling as graph problems and is known as sequential methods. The second category models the timetabling as a group of events (variables) and assigns their values (resources such as rooms and teachers) to achieve several soft and hard constraints. It is known as the constraint-based method [6]. In the third category, which is called the cluster method, the problem is partitioned into several event sets. The last category is called the meta-heuristic method, which includes different algorithms such as Ant Colony Optimization (ACO), neural networks, tabu search, simulated annealing, and genetic algorithms [7].

Besides, many innovative approaches were recently developed for solving timetabling problems, which involve hyper-heuristics, fuzzy, and case-based reasoning approaches [8].

In course timetabling, the enrolment-based problem is focused, which can be partitioned into six approaches: hyper-heuristic, population-based, hybridization meta-heuristic, meta-heuristic, graph-based, and constraint-based approaches.

In practice, meta-heuristics have recently achieved great usefulness in almost solving timetabling problems. The capability of escaping from local optima and handling a wide range of constraints is the key benefit of such methods [3].

To work out timetable problems of a Spanish secondary school [9], an approach of two stages is introduced. In the first stage, a heuristic algorithm is employed to produce the early answer. Improving this answer is done in the next stage using a tabu search algorithm. The Hyper-heuristic approach is applied in another work [10] to move up the generalization level of the computerized timetabling systems using a tabu search algorithm. The system was established to solve the problems of nurse scheduling and to achieve the course timetabling instances. Moreover, the course timetabling problem of enrollment-based was also solved using the tabu-based memetic algorithm, as introduced in ref. [11]. Incorporating a tabu list enhanced the effectiveness of the solutions and controlling the selections of neighborhood structures using the memetic algorithm [12] presented the tabu Search algorithm as a solution to a practical problem of university timetabling. Nine practical cases have been tested in that work. Further work focused on the lecture lengths at the University of Kebangsaan Malaysia, Faculty of Engineering, to resolve the university course timetable, using multi-neighborhood structures and tabu search [13]. In contrast, the winner of the 2004 International Timetabling Competition applied an approach of three stages to solve problems of course timetabling. Out of 20 examples in the competition, the approach has achieved 13 best results.

Feasible solutions, as an enhancement stage for the problem of course timetabling, were obtained [14], using a simulated annealing algorithm. Two types of neighborhood structures were the basis of this enhancement. In another work [15], to answer the post-enrollment course timetabling (PE-CTT), a simulated annealing algorithm was also applied within a metaheuristic approach. As compared to previous works, this work showed improved results regardless of the fact of using a somewhat straightforward single-step algorithm, against using complicated multi-step algorithms of most of the previous solvers. For complicated cases of university course timetabling, [16] introduced a non-linear great deluge algorithm. It is an extended version of the traditional great deluge algorithm. In contrast, a hybrid of Kempe-chain neighborhood structure with a great delug algorithm [17] was introduced to resolve the timetabling problem. This approach can compete with other available approaches. Afterwards, the great deluge algorithm is combined with an electromagnetic-like mechanism to obtain a new meta-heuristic [18]. The most important goal of the hyper-heuristic approach is to build up a general optimization approach that can be reused in diverse problem cases as well as a range of various problems. Lastly, a novel method was introduced for answering the problems of university course timetabling, which consists of two ant colony systems with hybrids. The first system combined the ant colony system and a simulated annealing algorithm, while the second system combined the ant colony system with a tabu search algorithm [19]. In the literature, Exact algorithms and approximation algorithms are the two types of algorithms that are utilized to solve the timetabling problem. In truth, precise algorithms can uncover the best answers; on the other hand, in the scheduling problem, the exact algorithms frame brute-force style. All these issues resulted in an exponential growth rate in the search space; therefore, these kinds of algorithms could only apply for the small in size issues. However, approximation algorithms are based on optimal solutions or not; nevertheless, they can make a good practical solution. As a result of all these factors, we proposed algorithms to solve the educational institution timetabling issue [20].

3 Describing the problem

Timetabling problem can be described by assigning course materials to a collection of rooms and timeslots, taking into account a group of soft constraints and hard constraints. The problem involves two stages to obtain the final solution. The first stage is to attain a feasible solution depending upon satisfying the hard constraints. The second stage is to obtain a better solution by improving the primary solution through a reduction in the soft constraints as much as possible. Thus, the final solution can compete with other relevant solutions in the literature [5].

We introduced a set of problem cases, which are used in this paper to test our algorithm. First, the hard constraints are listed in the following:

- At any time, assigning only one course to the student.
- The room should achieve the course feature requirements.
- No more students can attend the course greater than the room capacity.
- For each room, assigning only one course is to the same timeslot.

Second, the soft constraints are listed as follows:

- The last day-timeslot should have a course scheduled for the student.
- At least two consecutive courses should assign to the student.
- On a day, only one course should the student have.

Lastly, the problem includes the following:

- No. of courses = N, $e = \{e_1,\ldots,e_n\}$.
- No. of timeslots = 45.
- No. of rooms = R.
- No. of room features = F.
- No. of students = M.

The final solution (the aim of the problem) is to achieve a feasible timetable (the hard constraints) with minimizing the soft-constraints infringement.

This study is based on an instance of PE-CTT difficulties that was presented. Large, medium, and small datasets are divided into three categories based on their size. Eleven case studies are also taken into account: 5 mediums, 1 large, and 5 small (see Section 5).

4 The algorithm

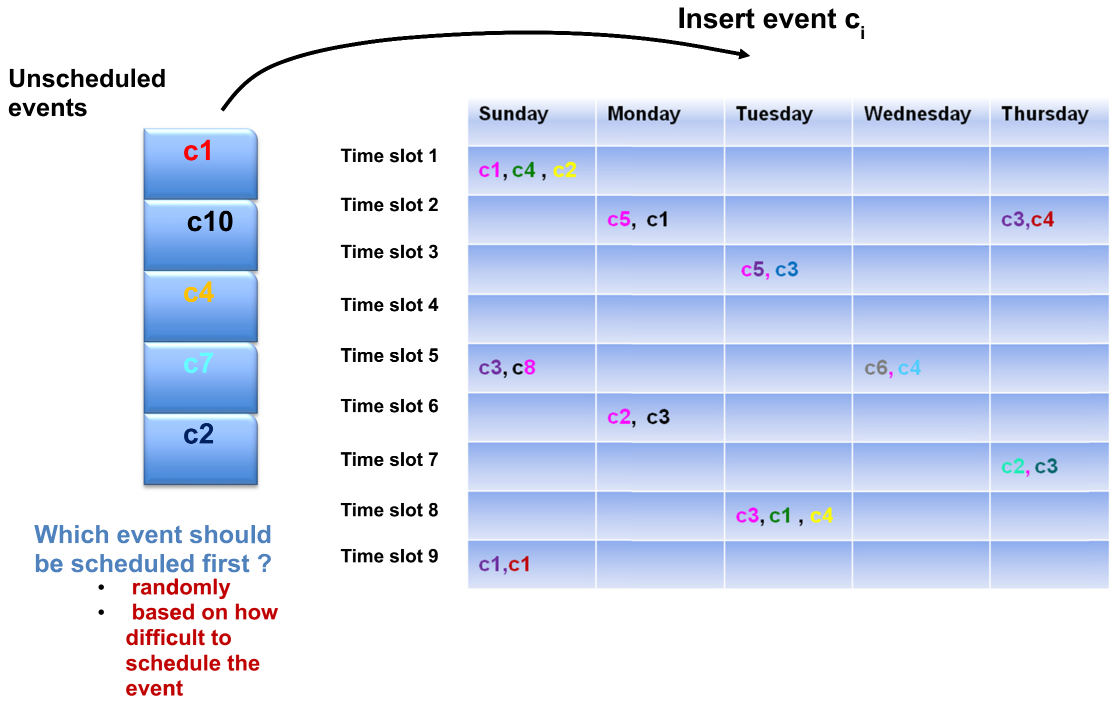
The work is partitioned into constructing and improving algorithms. The constructing algorithm is performed using the least saturation degree algorithm to get an early solution. Next, in the improving algorithm, the tabu search algorithm is employed to decrease the soft-constraints infringement (improvement).

The following four neighborhood structures are utilized:

1. **Nb1**: Randomly choose a particular course and progress to a feasible timeslot, which can produce the smallest cost.
2. **Nb2**: Randomly select a room. Also, randomly select two courses for that room. Next, swap timeslots.
3. **Nb3**: Randomly select two times. Next, swap timeslots.
4. **Nb4**: Randomly select a particular time and swap it with another time, in the range between 0 and 44, which can produce the smallest penalty cost.

4.1 Constructing algorithm

The first stage of the work includes the production of the early solution, which can achieve the hard constraints without any accounting for the soft constraints infringement. A least saturation degree algorithm is employed in this stage. First, the events beside smaller number of available rooms that seems hardly scheduled are selected. The algorithm is terminated if a feasible solution is obtained. Else, the


Insert event c_i
Unscheduled events
c1
c10
c4
c7
c2
Which event should be scheduled first ?
• randomly
• based on how difficult to schedule the event

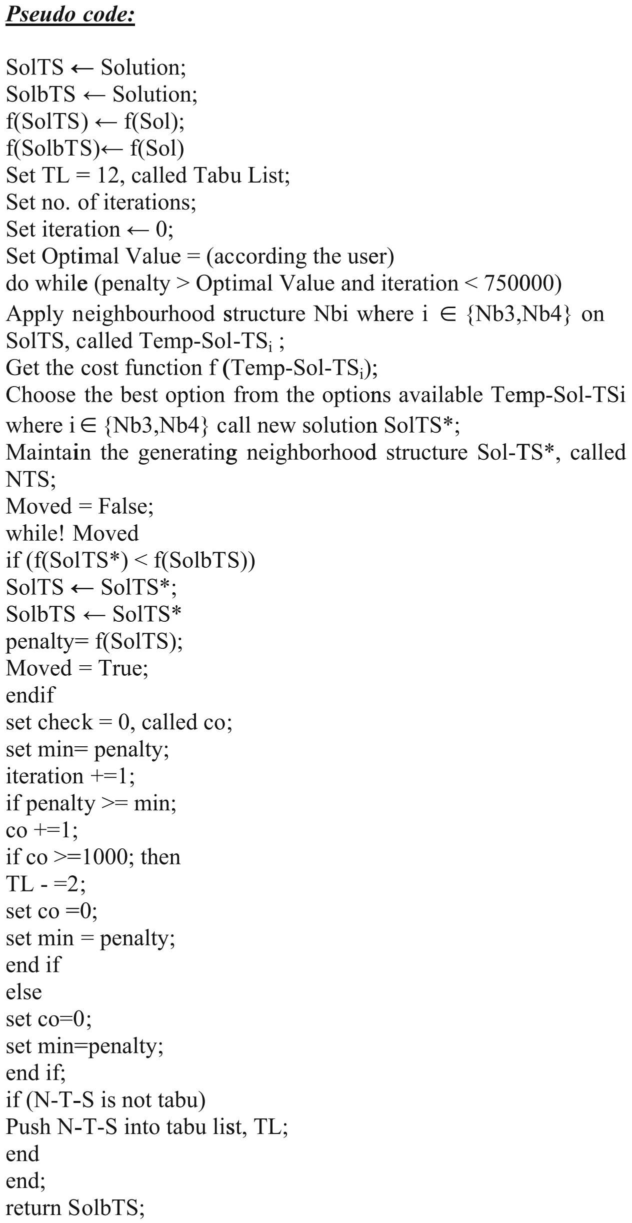
	Sunday	Monday	Tuesday	Wednesday	Thursday
Time slot 1	c1, c4 , c2				
Time slot 2		c5, c1			c3,c4
Time slot 3			c5, c3		
Time slot 4					
Time slot 5	c3, c8			c6, c4	
Time slot 6		c2, c3			
Time slot 7					c2, c3
Time slot 8			c3, c1 , c4		
Time slot 9	c1,c1				

Figure 1: Constructive heuristics (the least saturation degree).

neighborhood moves (N1 and/or N2) are performed to achieve feasibility. N1 is employed for a specific number of repetitions. Next, the algorithm is terminated if a feasible solution is achieved. Else, the algorithm continues using the N2 neighborhood structure for a specific number of repetitions. Note that all case studies are examined before the improvement algorithm is employed since the solutions must be made feasible first (see in Figure 1).

4.2 Improvement algorithm

The second stage includes using a set of neighborhood structures that were previously defined (Nb3, Nb4). During this stage, no hard-constraint infringement is completely allowed. Adaptive Tabu Search algorithm is applied to improve the early solution. Figure 2 shows the pseudocode of the used algorithm.



Pseudo code:

```
SolTS ← Solution;
SolbTS ← Solution;
f(SolTS) ← f(Sol);
f(SolbTS)← f(Sol)
Set TL = 12, called Tabu List;
Set no. of iterations;
Set iteration ← 0;
Set Optimal Value = (according the user)
do while (penalty > Optimal Value and iteration < 750000)
Apply neighbourhood structure Nbi where i ∈ {Nb3,Nb4} on
SolTS, called Temp-Sol-TS$_i$ ;
Get the cost function f (Temp-Sol-TS$_i$);
Choose the best option from the options available Temp-Sol-TSi
where i ∈ {Nb3,Nb4} call new solution SolTS*;
Maintain the generating neighborhood structure Sol-TS*, called
NTS;
Moved = False;
while! Moved
if (f(SolTS*) < f(SolbTS))
SolTS ← SolTS*;
SolbTS ← SolTS*
penalty= f(SolTS);
Moved = True;
endif
set check = 0, called co;
set min= penalty;
iteration +=1;
if penalty >= min;
co +=1;
if co >=1000; then
TL - =2;
set co =0;
set min = penalty;
end if
else
set co=0;
set min=penalty;
end if;
if (N-T-S is not tabu)
Push N-T-S into tabu list, TL;
end
end;
return SolbTS;
```

Figure 2: Pseudocode for the adaptive tabu search algorithm.

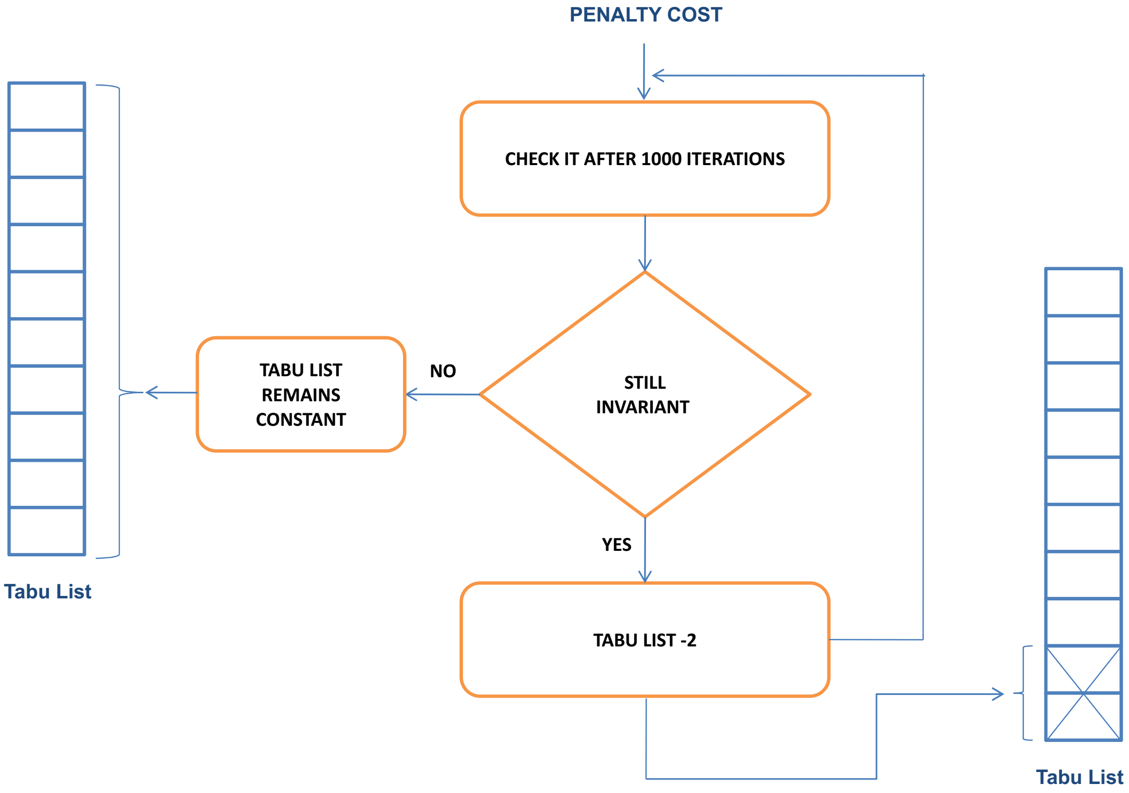


Figure 3: Adaptive tabu list.

Adaptive tabu search depends on the tabu list and the penalty cost, by checking the changing of this penalty cost, if this changing is still invariant for a period of iterations (1,000 iterations), the tabu list is reducing automatically (−2); otherwise, the tabu list remains constant. This procedure will continue until the tabu list equals 2, Figure 3.

5 Experimental results

Our algorithm is programmed in VB.NET using a PC of Win. 7, 1.7 GHz CPU, and 2G RAM.

Datasets are used for evaluating our algorithm efficiency. This collection of data was composed of different real-world course timetabling issues at different universities. We measured the quality of the timetable in the same way that did, by assigning a “1” to each violation of the soft or hard limits for each student involved. The hard constraint violations hvc (total penalty of hvc) of S* for all pupils, as well as the soft constraint violations svc (total penalty of svc) of S* for all students, make up the nature of the candidate solution S*, f(S*) (S equal total number of students), The datasets include 1 large, 5 mediums, 5 small instances that are available at http://iridia.ulb.ac.be/~msampels/tt.data/.

Table 1: The parameter values for the different types of course scheduling problems

Type	Small	Medium	Large
No. of courses	100	400	400
No. of rooms	5	10	10
No. of features	5	5	10
No. of students	80	200	400

Table 2: Compare our results with the results of other tabu search

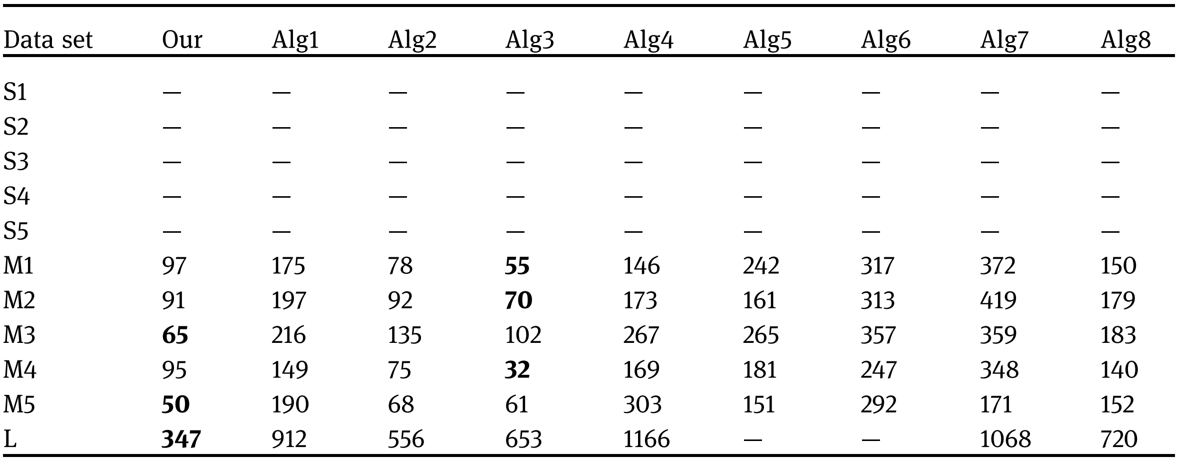
Data set	Our	Alg1	Alg2	Alg3	Alg4	Alg5	Alg6	Alg7
S1	–	–	–	–	–	–	–	–
S2	–	–	–	–	–	–	–	–
S3	–	–	–	–	–	–	–	–
S4	–	–	–	–	–	–	–	–
S5	–	–	–	–	–	–	–	–
M1	97	254	174	227	139	175	143	124
M2	91	258	184	180	92	197	130	117
M3	65	251	188	235	122	216	183	190
M4	95	321	180	142	98	149	133	132
M5	50	276	132	200	116	190	169	73
L	347	1072	994	–	615	912	825	424

Table 3: Comparing our results with other approaches in the literature

Data set	Alg8	Alg9	Alg10	Alg11	Alg12	Alg13	Alg14
S1	–	–	–	–	–	–	–
S2	–	–	–	–	–	–	–
S3	–	–	–	–	–	–	–
S4	–	–	–	–	–	–	–
S5	–	–	–	–	–	–	–
M1	180	140	117	9	338	236	243
M2	176	130	121	15	326	158	325
M3	219	189	158	36	384	261	249
M4	150	112	124	12	299	176	285
M5	196	141	134	3	307	147	132
L	–	876	645	208	100% Inf	296	1138

The experiments for the course timetabling problem discussed in this paper were tested on the benchmark course timetabling problems that need to schedule (100–400) courses into a timetable with (45) timeslots corresponding to (5) days of (9) hours each, while the room features satisfying and room capacity constraints also satisfying. Table 1 lists the parameter values that define the types.

On the eleven timetabling instances, we compare our technique to other algorithms in two parts: first, comparing with other literatures that used tabu search even if it hybrid with another algorithm (Table 2), and second, comparing with other works that used another algorithm (Table 3).



Data set	Our	Alg1	Alg2	Alg3	Alg4	Alg5	Alg6	Alg7	Alg8
S1	–	–	–	–	–	–	–	–	–
S2	–	–	–	–	–	–	–	–	–
S3	–	–	–	–	–	–	–	–	–
S4	–	–	–	–	–	–	–	–	–
S5	–	–	–	–	–	–	–	–	–
M1	97	175	78	**55**	146	242	317	372	150
M2	91	197	92	**70**	173	161	313	419	179
M3	**65**	216	135	102	267	265	357	359	183
M4	95	149	75	**32**	169	181	247	348	140
M5	**50**	190	68	61	303	151	292	171	152
L	**347**	912	556	653	1166	–	–	1068	720

Alg1: Abdullah et al. (2012) combine an electromagnetic-like mechanism (EM) and the great deluge algorithm (GD).

Alg2: Shaker (2010) Great Deluge and *Tabu Search*.

Alg3: Hamza & Salwani (2009) Incorporating Tabu Search into Memetic Approach.

Alg4: Burke et al. (2003a) a hyper-heuristic used tabu search.

Alg5: Abdullah, Burke and McCollum composite neighborhood structure with a randomized iterative improvement algorithm.

Alg6: Abdullah et al. (2007b) Variable neighborhood search with tabu.

Alg7: Burke et al. (2007).

Alg8: Ayob and Jaradat (2009) hybrids Ant Colony Systems with the tabu search (ACS-TS).

Alg1: Salwani and Hamza (2008) Genetic Algorithms and Local Search.

Alg2: AI-Betar et al. (2012) A MultiSwap Algorithm.

Alg3: Jat and Yang (2008) a memetic with genetic algorithm.

Alg4: Yang and Jat (2011) Genetic Algorithms With Guided and Local Search Strategies.

Alg5: Abdullah et al. (2010a) Hybridization algorithm. Electromagnetic-like mechanism with a great deluge.

Alg6: Abuhamdah and Ayob (2010) Round Robin (RR) Algorithm.

Alg7: Al-Betar et al. (2010) harmony search algorithm.

Alg8: Karami and Hasanzadeh (2012) hybrid genetic algorithm (HGA).

Alg9: Obit and Silva (2010) the non-linear great deluge Algorithm.

Alg10: Ayob and Jaradat (2009) hybrids Ant Colony Systems with the Simulated Annealing (ACS-SA).

Alg11: Sara et al. (2012) Simulated Annealing.

Alg12: Abdullah et al. Variable Neighborhood Search.

Alg13: Abdullah Hamdan (2008): A Hybrid Approach.

Alg14: Asmuni et al. Fuzzy Multiple Heuristic.

6 Conclusion

The aim of this paper was to develop a tabu search algorithm to contain an adaptive tabu list to be compatible with the state of solution for the course timetabling problem. This has been accomplished, demonstrating that tabu search is a viable solution to the course scheduling problem. In most cases, tabu search outperforms the traditional method. In addition, the usage of aspiration and stopping criteria are critical to the algorithm's effectiveness. Tabu search has been proved to be effective in solving scheduling issues, experimental results also showed that the most accurate timetables can be produced when hard and soft constraints are distinguished. Tabu search heuristics are also known to be sensitive to parameter selection. As a result, fine-tuning the parameters may yield even better outcomes. Future work includes further analysis of individual components of the contribution to develop a tabu search algorithm. Hence, future researchers can believe in limitations such as pre-requisite relations and the variation between the Compulsory and optional modules and solve this issue by promoting innovative algorithms, rather than the deep educational algorithm. In the same method, researchers are capable of compiling with other constraints to the model, or they can set heuristic algorithms. In future studies, the intended constraints are advised.

Conflict of interest: Authors state no conflict of interest.

References

[1] Osorio A, Esquivel M. A solution to the university course timetabling problem using a hybrid method based on genetic algorithms. Dyna. 2020;87(215):47–56.

[2] Amin R, Abshirini Z, Boshkani Zade M. Solving University course timetabling problem using parallel genetic algorithm. Int J Sci Res Comput Sci Eng. 2019;7.5:5–13.

[3] Khiarak NJ, Zamani-Harghalani Y, Derakhshi M-RF. Combined multi-agent method to control inter-department common events collision for university courses timetabling. J Intell Syst. 2020;29(1):110–26. doi: 10.1515/jisys-2017-0249.

[4] Abu Doush I, Al-Betar MA, Awadallah MA, Hammouri AI, Al-Khatib RM, ElMustafa S, et al. Harmony search algorithm for patient admission scheduling problem. J Intell Syst. 2020;29(1):540–53. doi: 10.1515/jisys-2018-0094.

[5] Wei M, Yang Y, Su J, Li Q, Liang Z. Task reallocating for responding to design change in complex product design. J Intell Syst. 2019;28(1):57–76. doi: 10.1515/jisys-2016-0262.

[6] Abdullah S, Turabieh H, McCollum B, McMullan P. A hybrid metaheuristic approach to the university course timetabling problem. J Heuristics. 2012;18:1–23. doi: 10.1007/s10732-010-9154-y.

[7] Bolaji AL, Okwonu FZ, Shola PB, Balogun BS, Adubisi OD. A modified binary pigeon-inspired algorithm for solving the multi-dimensional knapsack problem. J Intell Syst. 2021;30(1):90–103. doi: 10.1515/jisys-2018-0450.

[8] Turabieh H, Abdullah S. Incorporating tabu search into memetic approach for enrolment-based course timetabling problems. 2009 2nd Conference on Data Mining and Optimization. Selangor, Malaysia: IEEE; 27–28 October 2009. p. 115–9.

[9] Ayob M, Jaradat GH. Hybrid ant colony systems for course timetabling problems. 2nd Conference on Data Mining and Optimization. Selangor, Malaysia: IEEE; 27–28 October 2009. p. 120–6.

[10] Sadaf Jat N, Yang SH. A memetic algorithm for the university course timetabling problem. 20th IEEE International Conference on Tools with Artificial Intelligence; 2008. p. 427–33.

[11] Yang SH, Naseem Jat S. Genetic algorithms with guided and local search strategies for university course timetabling. IEEE Trans Systems, Man, Cybernetics-Part C: Appl Rev. January 2011;41(1):93–106.

[12] Jyoti M, Monga H, Baghla S. Reduction of inter-symbol interference using artifical neural network system in multicarrier OFDM system. Int J Wirel Microw Technol. 2018;8:10–8.

[13] Abuhamdah A, Ayob M. Adaptive randomized descent algorithm using round robin for solving course timetabling problems. 10th International Conference on Intelligent Systems Design and Applications, IEEE; 2010. p. 1201–6.

[14] Azmi Al-Betar M, Tajudin Khader A, Yi Liao I. A harmony search with multi-pitch adjusting rate for the university course timetabling. Berlin Heidelberg: Springer-Verlag; 2010. p. 147–61.

[15] Hossein Karami A, Hasanzadeh M. University course timetabling using a new hybrid genetic algorithm. 2nd International Conference on Computer and Knowledge Engineering (ICCKE); October 18–19 2012. p. 144–9.

[16] Henry Obit J, Landa-Silva D. Computational study of non-linear great deluge for university course timetabling. Berlin Heidelberg: Springer-Verlag; 2010. p. 309–28.

[17] Ghaith Jaradat M, Ayob M. Big bang-big crunch optimization algorithm to solve the course timetabling problem. 10th International Conference on Intelligent Systems Design and Applications. Selangor, Malaysia: Data Mining and Optimization Research Group The National University of Malaysia B.B.Bangi; p. 1448–52.

[18] Zheng N, Chen T, Lin F, Xu H. A hybrid heuristic algorithm for the intelligent transportation scheduling problem of the BRT system. J Intell Syst. 2015;24(4):437–48. doi: 10.1515/jisys-pp.2014-0134.

[19] Ceschia S, Di Gaspero L, Schaerf A. Design, engineering, and experimental analysis of a simulated annealing approach to the post-enrolment course timetabling problem. Comput Oper Res. 2012;39:1615–24.

[20] Alirezaei E, Vahedi Z, Ghaznavi-Ghoushchi M. Parallel hybrid meta heuristic algorithm for university course timetabling problem. 20th Iranian Conference on Electrical Engineering, (ICEE2012). Tehran, Iran: IEEE; May 15–17 2012. p. 673–8.