The effect of neighborhood structures on tabu search algorithm in solving university course timetabling problem

Ali Shakir, Belal AL-Khateeb, Khalid Shaker, and Hamid A. Jalab

Citation: AIP Conference Proceedings **1635**, 657 (2014); doi: 10.1063/1.4903652
View online: http://dx.doi.org/10.1063/1.4903652
View Table of Contents: http://scitation.aip.org/content/aip/proceeding/aipcp/1635?ver=pdfcov
Published by the AIP Publishing

The Effect of Neighborhood Structures on Tabu Search Algorithm in Solving University Course Timetabling Problem

Ali Shakir[a], Belal AL-Khateeb[b], Khalid Shaker[c] and Hamid A. Jalab[d]

[a,b,c]Computer Science Department, College of Computer, University of Anbar
Ramadi, Iraq.
alialani2010@yahoo.com
belal@computer-college.org
khalidalhity@gmail.com
[d]Multimedia Unit, Faculty of Computer Science and Information Technology,
University Malaya, 50603 Kuala Lumpur, Malaysia
hamidjalab@um.edu.my

Abstract. The design of course timetables for academic institutions is a very difficult job due to the huge number of possible feasible timetables with respect to the problem size. This process contains lots of constraints that must be taken into account and a large search space to be explored, even if the size of the problem input is not significantly large. Different heuristic approaches have been proposed in the literature in order to solve this kind of problem. One of the efficient solution methods for this problem is tabu search. Different neighborhood structures based on different types of move have been defined in studies using tabu search. In this paper, different neighborhood structures on the operation of tabu search are examined. The performance of different neighborhood structures is tested over eleven benchmark datasets. The obtained results of every neighborhood structures are compared with each other. Results obtained showed the disparity between each neighborhood structures and another in terms of penalty cost.

Keywords: Tabu Search, Meta-Heuristics, Course Timetabling problem, Neighborhood structure, Type of move**.**
PACS: 07.05.Bx, 07.05.Mh. 07.05.Tp

INTRODUCTION

The overall objective of the timetabling problem "is to assign a set of entities (such as tasks, public events, vehicles, or people) to limited number of resources over time, in such a way as to meet a set of pre-defined schedule requirements" [1]. In the university timetabling problem, "schedule requirements" are grouped into two generic constraints categories: 'hard' and 'soft' constraints. Satisfying hard constraints are necessary to produce a (feasible) timetable whilst soft constrains are desired but not absolutely essential. Soft constraints can be violated, as they are not essential but rather desirable. However, the more they are met, the more the solution gains optimality [2].

There are several ways suggested in several literatures in order to solve the problem of timetabling, each one used a certain way according to scientific specialization such as: Operations research, Artificial intelligence and computational intelligence .Various techniques have been applied to tackle university course timetabling problems i.e. Tabu Search [3], Ant Colony Optimization [4], Evolutionary Algorithm [5], Hybrid-Heuristic Algorithm [6], Simulated Annealing [7] and Hyper Heuristic approach [8].

The work done by Al_Tarawneh and Ayob [3], Aladag & Hocaoglu [9], Abdelkarim et al. [10], Nguyen et al. [11], Lü and Hao [12] and Tan and Thi [13] on Tabu Search algorithm is summarized below:

Al_Tarawneh and Ayob [3] apply a Tabu Search and multi-neighborhood structure to solve University Course Timetable at the faculty of engineering, University Kebangsan Malaysia, focusing on the length of lectures. Aladag and Hocaoglu [9] proposed a Tabu Search algorithm that applied to a timetabling problem of the Statistics Department of Hacettepe University. This work's given formulation does not contain conflicts in lessons of sections that was regarded as a soft constraint. Abdelkarim et al. [10] used tabu Search procedure for course timetabling at an institution in a Tunisian University. Nguyen et al. [11] applied Tabu Search algorithm to a real-world university timetabling problem in Vietnam. The algorithm is tested on nine real-world instances. Lü and Hao [12] present an Adaptive Tabu Search Algorithm (ATS). The proposed ATS algorithm integrates several features such as an original double Kempe chains neighborhood structure, a penalty-guided perturbation operator and an adaptive search mechanism. Tan and Thi [13] proposed the Bees algorithm to solve a real-world university timetabling problem in Vietnam.

There are other meta-heuristic approaches that have been used to solve course timetabling problem such as Shaker and Abdullah [14] used a hybrid of great deluge algorithm with kempe chain neighborhood structure to solve the problem of university course timetable. SolÍs et al. [15] applied a simulated annealing algorithm to get feasible solutions. Abdullah et al. [16] presented a new Meta heuristic that combines an electromagnetic-like mechanism (EM) and the great deluge algorithm (GD. Karami and Hasanzadeh [17] proposed a hybrid genetic algorithm (HGA) to improve local exploitation by using hill climbing algorithm. Jat and Yang [18] presents a memetic algorithm that integrates two local search methods into the genetic algorithm. Ayob and Jaradat [19] apply two hybrids Ant Colony Systems to solve the university course timetabling problem.

One of the most efficient algorithms for the solution of the problem of course timetabling is Tabu Search algorithm. Tabu Search technique is a meta-heuristic developed and independently in [20-24], has proved to be very efficient to solve many combinatorial problems and especially educational timetabling problems. The method was developed to overcome the previous local search methods that lead to local optima like hill climbing and descent methods. The originality of the procedure is the use of short-term memory in order to prevent the return to inverse moves (cycling) and long term memory in order to diversify and intensify the search space [10]. One of the most important factors which affect the efficiency of the algorithm is a defined neighborhood structure pertained to the nature of the problem [20].

In this paper, we examine different neighborhood structures based on types of move called algorithm. We apply five algorithms, each algorithm consist of many neighborhood structures. According to obtained results, multiple comparisons among all neighborhood structures are statistically done.

PROBLEM DESCRIPTION

The problem involves the assignment of course to timeslots and rooms subject to the set of hard and soft constraints. Hard constraints must be satisfied to get feasible solution then this solution is by reducing as much as possible the number of soft constraints to acquire a good solution that can compete with the other solutions in the relevant literature.

In this study, we examined our algorithm over the problem cases proposed in [22]; consequently the following hard constraints have been introduced: (1) It is highly impossible to assign the same student to more than one course simultaneously. (2) The room ought to fulfill the attributes demanded by the course. (3) It is essential that, number of students enrolled the course must be lesser than or equal to the room capacity. (4) It is highly impossible to allow more than one courses to be allocated to the same time-slot in each room.

The following soft constraints that are equally penalized have also been presented in [22]: (1) A student gets a course timetabled in the final time-slot of the day. (2) A student gets more than 2 continuous courses. (3) A student has a one course on a day.

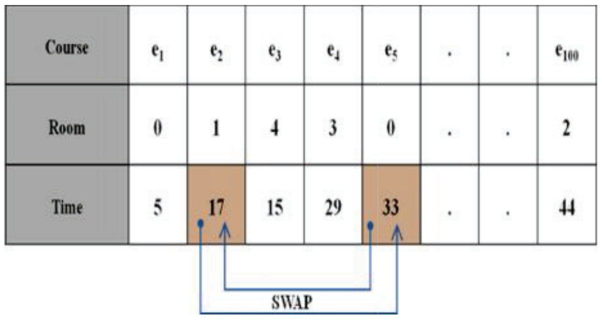
The aim is to fulfill the hard constraints (feasible timetable) and to reduce the breach of the soft constraints.

In this work, we consider post-enrollment course timetabling problems that proposed in [22] datasets are divided into three categories: small, medium and large. We deal with 11 instances: 5 small, 5 medium and 1 large (more details are in Experimental Results).

THE ALGORITHM

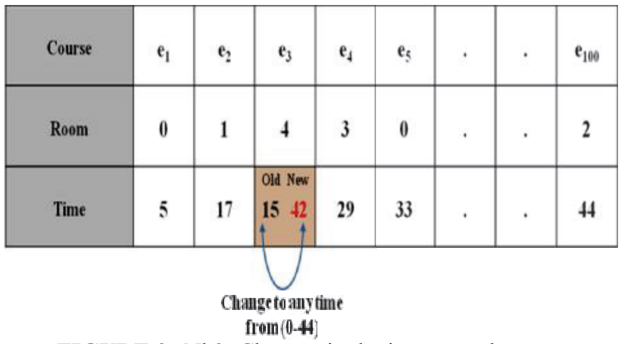
It is believed that due to hard constraints forced on the university course timetabling problem, the feasible region is highly scattered and sparse. So, the problem may fall in the local optima. To overcome this, a Tabu Search algorithm i.e. a memory-based meta-heuristic, which tries to escape from get bounds to local optima is applied [25, 26]. During this phase a set of neighborhood structures are applied to reduce the violation of soft constraints. These neighborhood structures are:

1. **Nb1**: Randomly choose two times-slots and exchange it (Fig.1).
2. **Nb2**: Choose a single time-slot randomly and change it with any time from (0-44) that can generate the lowest penalty cost (Fig.2).
3. **Nb3:** Select a one course randomly and swap it with another course for the same room (Fig.3).
4. **Nb4:** Randomly choose two courses from the same room (the room is arbitrarily chosen) and exchange time-slots (Fig.4).



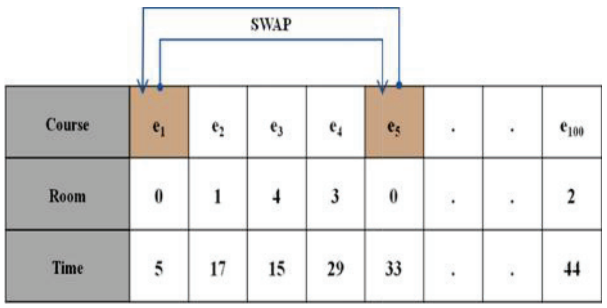
Course	e_1	e_2	e_3	e_4	e_5	.	.	e_{100}
Room	0	1	4	3	0	.	.	2
Time	5	17	15	29	33	.	.	44

FIGURE 1. Nb1: Swapping time randomly



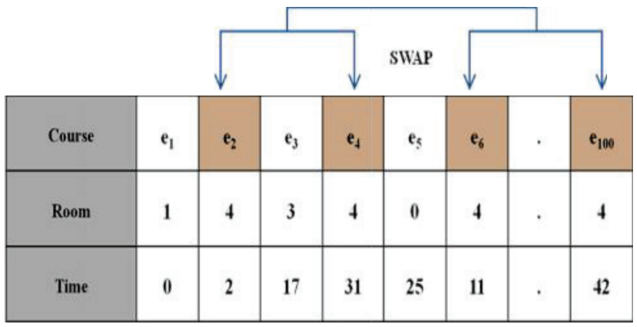
Course	e_1	e_2	e_3	e_4	e_5	.	.	e_{100}
Room	0	1	4	3	0	.	.	2
Time	5	17	Old New 15 42	29	33	.	.	44

FIGURE 2. Nb2: Change single time at random.



Course	e_1	e_2	e_3	e_4	e_5	.	.	e_{100}
Room	0	1	4	3	0	.	.	2
Time	5	17	15	29	33	.	.	44

FIGURE 3. Nb3: swapping one course with another for same room randomly



Course	e_1	e_2	e_3	e_4	e_5	e_6	.	e_{100}
Room	1	4	3	4	0	4	.	4
Time	0	2	17	31	25	11	.	42

FIGURE 4. Nb4: Swapping two courses with other for same room randomly

Improvement Algorithm

A set of the neighborhood structures outlined above are applied here. These neighborhood are used with different ways, These ways named as algorithm. We applied five algorithms as described below:

Algorithm1: Original Tabu search with two neighborhood structures:

Nb1: Randomly choose two times and exchange time- slots.

Nb2: Choose a single time randomly and change it with any time from (0-44) that can generate the lowest penalty cost.

Algorithm2: Original Tabu search with three neighborhood structures:

Nb1: Randomly choose two times and exchange time-slots.

Nb2: Choose a single time randomly and change it with any time from (0-44) that can generate the lowest penalty cost.

Nb3: Select a one course randomly and swap it with another course for same room.

Algorithm3: Original Tabu search with three neighborhood structures:

Nb1: Randomly choose two times and exchange time-slots.

Nb2: Choose a single time randomly and change it with any time from (0-44) that can generate the lowest penalty cost.

Nb4: Randomly choose two courses from the same room (the room is arbitrarily chosen) and exchange time-slots.

Algorithm4: Tabu search (decrement Tabu List) with two neighborhood structures:

Nb1: Randomly choose two times and exchange time-slots.

Nb2: Choose a single time randomly and change it with any time from (0-44) that can generate the lowest penalty cost.

Algorithm 5 (ATS): Adaptive Tabu search (ATS) – Fig.5 (decrement and increment Tabu List) with two neighborhood structures:

Nb1: Randomly choose two times and exchange time-slots.

Nb2: Choose a single time randomly and change it with any time from (0-44) that can generate the lowest penalty cost. The original tabu search is described in [27].

In this stage the hard constraints must be never violated while reducing the number of soft constraints is matter.

Fig.5, explains the three steps that must be implemented to get the best solutions (Initialization, Improvement which include (standard Tabu Search and Adaptation) and Stopping criterion):

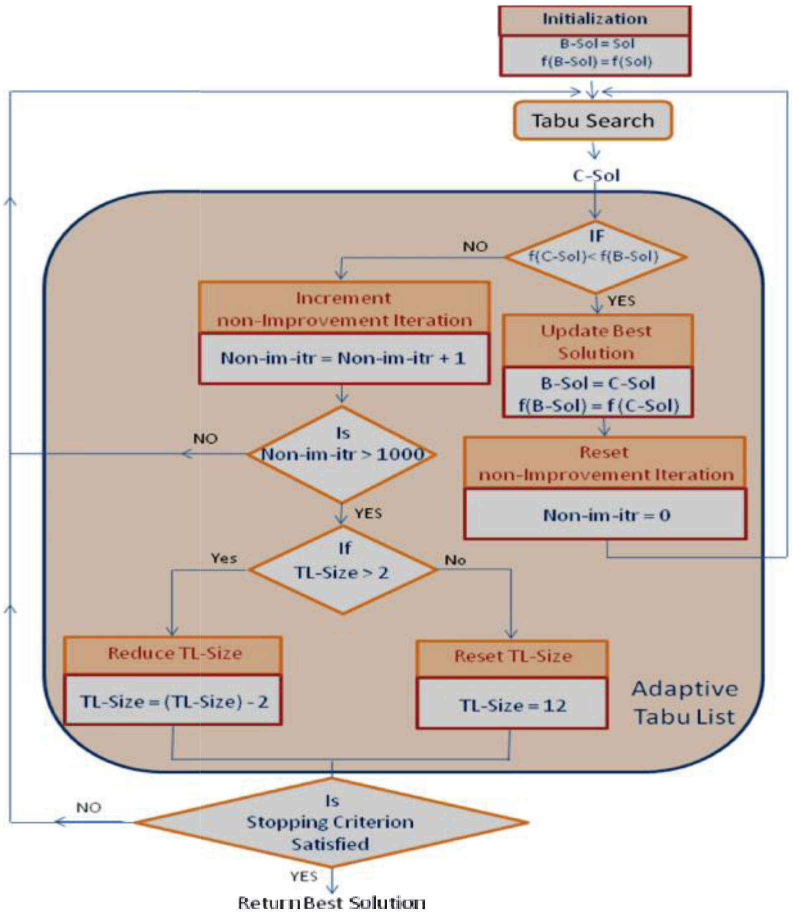


FIGURE 5. Steps of adaptive tabu search algorithm

- Step1: Initialization:

In this step the initial solution (Sol) and the cost function of the solution f(Sol) that satisfy all hard constraints are set to be (B-Sol) and f (B-Sol) respectively. Then these values will be entered to the second step.

- Step2: Improvement:

Step 2.1: Standard Tabu Search:

In this step the tabu search variables in the initialization phase are set as:

- The current solution B-Sol is set to be SolTS.
- Sets of neighborhood structures denoted by (N).
- Best solution denoted by SolbestTS.
- Current solution denoted by SolTS.

The termination criterion in this phase depends on the number of repetitions or the accomplishment of ideal value. In this experiment, we used two neighborhood structures (Nb1 and Nb2).

Step 2.2: Adaptation (Adaptive Tabu List)

In our work, the ATS lies in the adaptation of tabu list, which depend on the penalty cost and number of iterations. The initial solutions (B-Sol) enter to the tabu search two neighborhood structures (Nb1 and Nb2) are applied for many iterations to improve the penalty cost, as shown in Fig.5.

Step3: Stopping Criterion:

The termination criterion in this phase depends on the number of repetitions or the accomplishment of ideal value. The number of repetitions is set to be 450000 and was determined based on many experiments on different number of iterations.

EXPERIMENTAL RESULTS

Our algorithm is implemented using Visual Besic and tested on a PC running Windows 7 with Intel Core i7-2600K and 16GB RAM. In order to gauge the performance of all neighborhood structures, timetables are generated by least saturation degree algorithm to generate the initial solutions that will be improved later.

To evaluate the efficiency of our work, we tested the experiments in [22] data sets i.e (5 small, 5 medium and 1 large instances) which are available at http://iridia.ulb.ac.be/~msampels/tt.data/.

The experiments of course timetabling problem presented in this paper have been examined on the benchmark course timetabling problems, which have to allocate (100-400) courses into a timetable with (45) time-slots, equivalent to five days of nine hours each, while fulfilling the limitations of room attributes and capacity.

In this test, we have assessed the search prospective of our algorithm using a stopping criterion. For this purpose, as we said the algorithm was run for 450000 iterations with different set of moves. The best results out of 11 runs are presented. Table (1) shows the comparison of our algorithms.

TABLE 1. Penalty cost results of all algorithms

Dataset	Alg1	Alg2	Alg3	Alg4	ATS
S1	0	9	10	0	0
S2	0	5	5	0	0
S3	0	4	5	0	0
S4	0	1	1	0	0
S5	0	10	12	0	0
M1	105	150	170	97	95
M2	100	111	119	91	90
M3	75	89	102	65	63
M4	96	110	109	95	94
M5	59	81	100	50	50
L	435	473	483	405	405

The table above explains the use of different neighborhood structures with (original and adaptive) tabu search. The results indicate that ATS (decrement and increment tabu list) gives better solutions in four instances (M1,M2,M3 and M4) comparing with all other algorithms. The results show that ATS and algorithm 4 are of equal performances on the remaining instances (M5 and Large). Fig. 6 through 11, show the differentiation between these algorithms in terms of penalty cost. Those figures proved that ATS has the best obtained results in all instances; Except for instances (M5 and L) at which both ATS and algorithm 4 are the same.

Fig. 6-11, show the frequency charts of the neighborhood structures that have been used for datasets (M1, M2, M3, M4, M5 and L). The x-axis represents the algorithms used with the neighborhood structures employed throughout the search, while the y-axis represents the values of penalty cost. It is clear that in most of the medium and large instances, neighborhood structures move1 (Nb1) and move2 (Nb2) are the most popular structures used and gives best solutions (Alg1,Alg4 and ATS), especially if they are used together. It is also clear that some neighborhood structures do not contribute to a good quality solution as in (Alg2 and Alg3), which used (Nb3) and (Nb4).

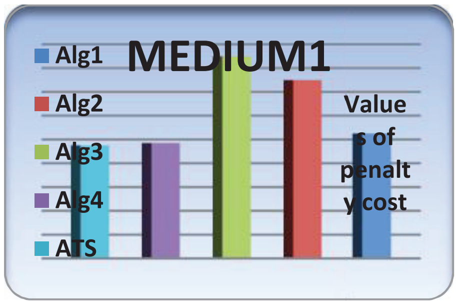


FIGURE 6. Frequency of the neighborhood structures used for Medium 1

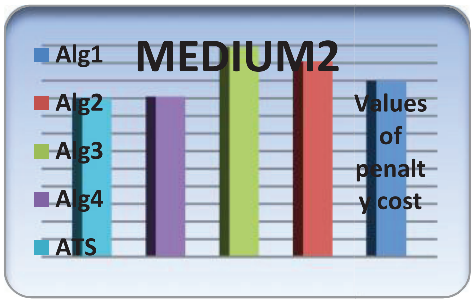


FIGURE 7. Frequency of the neighborhood structures used for Medium 2

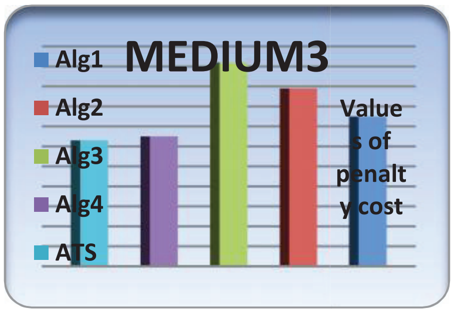


FIGURE 8. Frequency of the neighborhood structures used for Medium 3

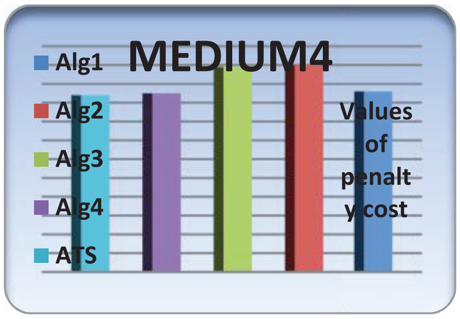


FIGURE 9. Frequency of the neighborhood structures used for Medium 4

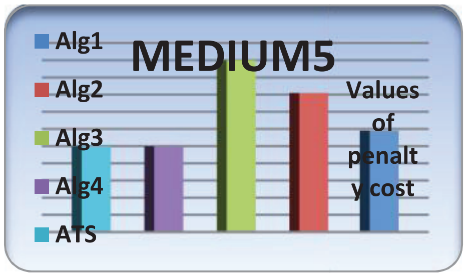


FIGURE 10. Frequency of the neighborhood structures used for Medium 5

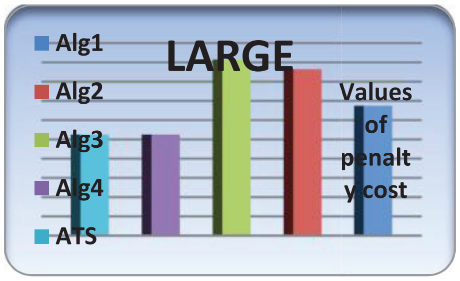


FIGURE 11. Frequency of the neighborhood structures used for Large

The preliminary results show that good moves (regardless of the number of moves) will help the algorithm to find better solutions. This shows the advantage of combining several good neighborhood structures against the type of structure alone in order to help compensate against the ineffectiveness of the other neighborhood structures.

Fig. 12- 15, show the box plot that summarizes the results of 11 runs for each dataset (five medium and one large) by all algorithms in [22].

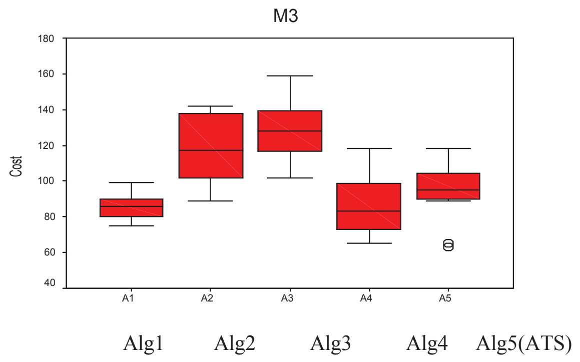


FIGURE 12. Box plots of the penalty costs for medium and large datasets.

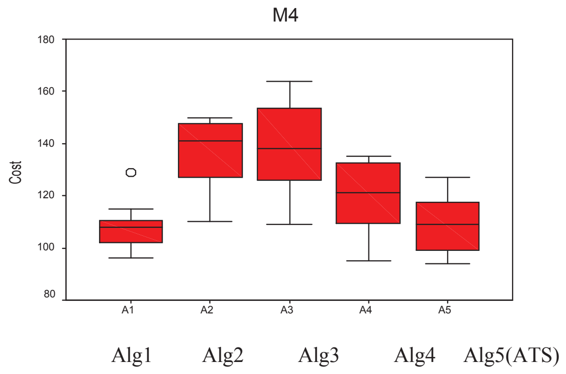


FIGURE 13. Box plots of the penalty costs for medium and large datasets.

The outcomes illustrated in figures depict lesser distribution of solution points for M3 and M5 instances for Alg1, Alg4 and Alg5 (ATS). It is of our opinion that, the neighborhood structures (Nb1 and Nb2) employed together to these datasets are capable of forcing the search algorithm to broaden its pursuit of the solution space, by shifting from one neighborhood structure to the other, despite the fact that there might be much less and more scarcely dispersed solution points in the solutions space, considering that a lot of courses are conflicting with each other.

Based on these experiments, we consider that, the dimensions of the search space might not be reliant on the size of problem, because the distribution of solution points significantly varies from one to another, although the issues are from the same group of datasets with similar parameter values.

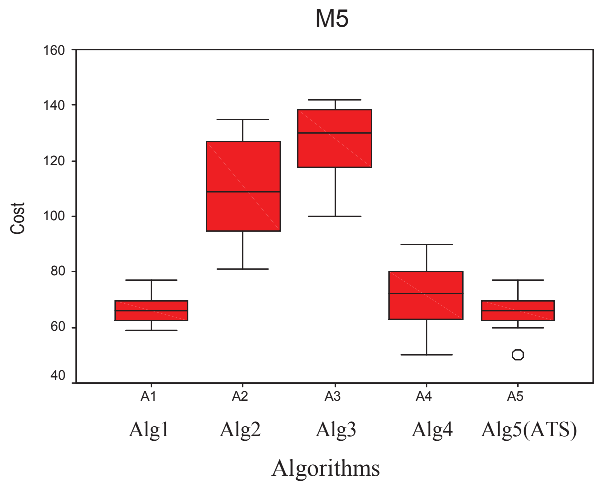


FIGURE 14. Box plots of the penalty costs for medium and large datasets.

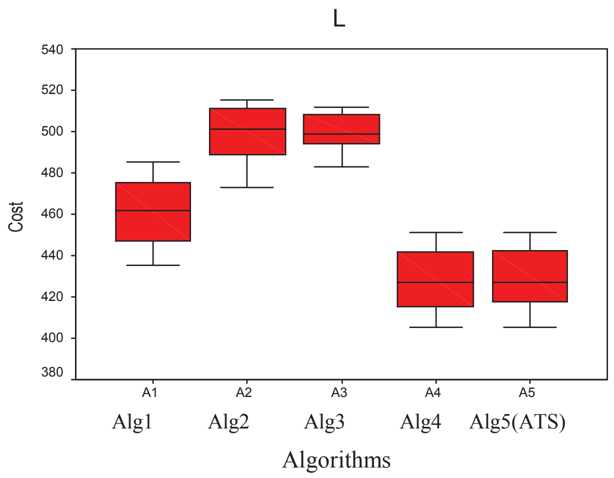


FIGURE 15. Box plots of the penalty costs for medium and large datasets.

CONCLUSIONS AND FUTURE WORKS

Four neighborhood structures based on simple and swap moves were used in this paper. These moves affect the Tabu Algorithm in different aspects. The simple move causes more intensification effect since it alters just one time. On the other hand, the swap move causes more diversification effect since it interchanges two courses or two times-slots. The course timetabling problem is solved by using these all structures. Also, initial solutions are generated for the problem. Then, the results are statistically examined .

The performance of these neighborhood structures was compared against each other, the results obtained proved that the best timetables are obtained using (Nb1 and Nb2).

Furthermore, future studies should consider including additional limitations (i.e. soft constraints), and aim to validate this algorithm on the International Timetabling Competition datasets (ITC-2007), and as well as on real world timetabling problems.

REFERENCES

1. R. Lewis, *OR Spect*, **30**, 167-190 (2008).
2. M. AI-Betar, in: Proceeding of ICCI (2012), pp. 301-306.
3. T. Younis and A. Masri, in*: Proceeding of DMO* (2011), pp. 208-212.
4. K. Socha, in: Proceeding of EWECCO (2003), pp. 334-345.
5. O. Rossi-Doria, M. Samples, M. Birattari, M. Chiarandini, M. Dorigo, L. M. Gambardella, J. Knowels, M. Manfrin, M. Mastrolilli, B. Paechter, L. Paquete, and T. Stultzle, in: *Practice and Theory of Automated Timetabling VI*, (Springer, Heidelberg, 2003), pp. 329–354.
6. P. Kostuch, in: *Practice and Theory of Automated Timetabling V*, (Springer, Heidelberg, 2005), pp. 109-125.
7. J. Frausto-Solís, F. Alonso, and J. Mora-Vargas, in: *Proceeding of MICAI* (2008), pp. 675–685.

8. R. Bai, E. K. Burke, G. Kendall, and B. McCollum, in*: Proceeding of PATAT'06* (2006), pp. 301-306.
9. A. Hakan and H. Äum*, Hacet J. Math. Statis*, **36**,1 (2007).
10. E. Abdelkarim, K. Hichem, and J. Dammak, in*: Proceeding of ICPTAT(2008),* pp. 150-164.
11. N. Khang, D. Nguyen, T. Khon & T. Nuong, in*: Proceeding of RIVF(2010),* pp. 1-6.
12. L. Zhipeng and H. Kao, *Eur. J. OR,* **200**, 235(2010).
13. K. Nguyen and N. Tran, in*: Proceeding of KSE* (2011), pp. 301-306.
14. S. Khalid and A. Salwani, in*: Proceeding of CDMO (*2009), pp. 149-153.
15. S. Frausto, J., Alonso-Pecina, F. & V. Mora, in*: Proceeding of MICAI* (2008), pp.675-685.
16. A. Salwani, T. Hamza, M. C. Barry & M. M. Paul*,* J *Heuristics,* **18**,1(2012).
17. K. A. Hossein and H. Maryam, in*: Proceeding of ICCKE* (2012), pp.144-149.
18. N. Sadaf Jat and Y. Shengxiang, in*: Proceeding of ICTAI* (2008), pp. 427-433.
19. A. Masri and J. Ghaith, in*: Proceeding of DMO* (2009), pp. 120-126.
20. C. H. Aladag and G. Hocaoglu, in*: Proceeding of STC* (2007), pp. 14–19.
21. M. Carter and G. Laporte, in*: Proceeding of ICPTAT*(1998), pp. 3–19.
22. K. Socha, J. Knowles and M. Samples, in*: Proceeding of ANTS* (2002), pp. 1-13.
23. F. Glover, *Comp.OR*, **13**, 533(1986).
24. P. Hansen, in*: Proceeding of CNMCO* (1986), pp. 55-69.
25. F. Glover and M. Laguna, in: *Tabu search*, (Kluwer, Dordrecht, 1997).
26. S. Khalid, "An Investigation of Meta-Heuristic Algorithms for University Course Timetabling Problems"Ph.D Thesis, UKM, 2010.
27. H. Hoos, *Stochastic Local Search Foundations and Applications*, (Morgan Kaufmann, 2005).