Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 161 (2019) 656–664

Procedia Computer Science

www.elsevier.com/locate/procedia

The Fifth Information Systems International Conference 2019

Automated Course Timetabling Optimization Using Tabu-Variable Neighborhood Search Based Hyper-Heuristic Algorithm

Ahmad Muklason*, Redian Galih Irianti, Ahsanul Marom

Department of Information Systems, Institut Teknologi Sepuluh Nopember, Surabaya

Abstract

Course timetabling problems are challenging, laborious and repetitive work in the universities. However, in many universities this repetitive works was still carried out manually. It is because there are so many constraints that must be considered either from the students and lecturers' requirement or the infrastructure such as room availability. Therefore, to automate the process of timetabling is hard problem. Scientifically, in the literature, course timetabling optimization is one of non-deterministic polynomial problems, usually abbreviated as NP-hard problem. For NP-hard problem, there is not any exact algorithm known could solve the problem within polynomial-time. The state-of-the-art methods for solving the problem are approximation algorithms that are mainly meta-heuristics. This paper presents a new approach, namely, hyper-heuristics as opposed to meta-heuristics, to cope with the need of intensive problem specific parameter tuning in meta-heuristics approach. The algorithms employed within hyper-heuristic approach presented in this paper are tabu search hybridize with variable neighborhood search. Tested over two real-world course timetabling problem datasets, the computational results from the experiments showed that the proposed algorithm could automate the process of timetabling. Furthermore, compared to the timetable produced manually, in term of soft constraint violation penalties, the proposed algorithm could improve by 1855 and 1110 respectively. In addition to new approach, the main contribution of this paper is two real-world course timetabling problems available for public to encourage further research as future works.

Keywords: Course Timetabling; Variable Neighborhood Search Algorithm; Tabu Search Algorithm; Hyper-Heuristic Algorithm

* Corresponding author. Tel.: +62-812-2825-6102.
E-mail address: ahmad.muklason@gmail.com

1877-0509
10.1016/j.procs.2019.11.169

1. Introduction

Manual scheduling still causes recurring problems when making a schedule. These problems include the time allocated or needed to make a schedule is still time consuming. Manual course scheduling can take up to 2 weeks, although scheduling is a routine activity that must be undertaken each semester. The course schedule of manually generated is also less flexible. This will be difficult when sudden changes occur, such as when there is a request from a lecturer. Other problems that occur come from the student. There are still students who get a very dense course schedule in a day. This certainly makes students bored to take courses so that it also influences the performance of students in doing assignments in each subject. The course scheduling manually is still considered to be not optimal due to students who get a full schedule in a day, while on the following day, they off. It would be better if students get courses that don't overlap on a day.

Based on the some above mentioned problems, optimization in course timetabling is required. Several related approaches and methods have been developed. One of them is hyper-heuristic by combining the Variable Neighborhood Search algorithm and Tabu Search, that will be used in this research. Tabu-Variable Neighborhood Search was chosen because it has been proven effective to solve vehicle routing problem in [1]. Therefore, it is interesting to know whether the algorithm effective to solve another combinatorial optimization problem, in this case course timetabling problem.

Hopefully, the optimization of the course scheduling can provide an optimal courses schedule for students and have flexibility to change. The purpose of the course scheduling optimization is to minimize the time of scheduling courses needed and the number of courses in the same semester in a day. The goal is achieved by still considering several constraints, namely the number of subjects, the allocation of rooms, the number of students, the number of lecturers, and the allocation of lecture time.

The rest of this paper is organized as follows. Section 2 describes the related work and literature studies that supports this research followed by the problem formulation in Section 3. Section 4 explains the methodology how to implements Tabu-Variable Neighborhood Search. Experimental result and discussion of implementation of Tabu-Variable Neighborhood Search based Hyper-Heuristics are presented in Section 5. Finally, conclusion and future work are presented in Section 6.

2. Related works

This section explains the literature studies and some related works that support this research.

2.1. Timetabling problem

Timetabling problem is an activity that is mostly done at educational institutions or other institutions. Some problems often occur in the scheduling process. One problem is that it takes too much time to make a schedule manually. Scheduling can be defined as a problem that has four parameters: a set of time (T), a set of resources (R), a set of meetings (M), a set of constraints (C). In this case, the problem in question is related to allocate time and resources to a meeting to meet as many constraints as possible [2].

The course timetabling problem is a problem that appears periodically at a college. Common problems consist of setting agendas such as classes, lectures, tutorials, etc. to become several time slots which meet any constraints. The constraints must be met are hard constraints and soft constraints. Models are made may not violate existing hard constraints. While, the results of the implementation will be better when it can meet the soft constraints, but it can still be accepted with penalties [3]. This research uses hard constraint:

- All events must be scheduled.
- Events taken by the same student cannot be scheduled at the same timeslot.
- The number of students taking an event must not exceed the capacity of the room where the event is scheduled.
- The Timeslot and the rooms cannot be used more than once.

Soft constraints are defined as limits which can be violated but the amount must be minimized, these limitations include: (i) Students may not take more than 2 consecutive time slots for on a day. (ii) Students may not take only 1 course on a day. (iii) Students may not take course at the last time slot on a day.

In addition to course timetabling problem, other common educational timetabling problems are examination timetabling problem [4] and school timetabling problem [5]. More recent works on examination timetabling problem can be observed in [6,7,8].

2.2. Hyper-heuristics

Hyper-heuristic is a high-level search methodology that searches over heuristic spaces. The basic idea of a hyper-heuristic algorithm is combining different heuristics to harness the power of each heuristic since the 1960s. Since then, interest in hyper-heuristics has increased. A theoretical study shows that combining heuristics can lead to exponentially faster search compared to using independent heuristics in some cases [9].

A heuristic choice generally combines heuristic choices then moves the acceptance method under an iterative framework. For each step, low level heuristic is used to modify the existing solution [9]. In addition to course timetabling problems, Hyper-heuristic approaches have been successfully employed for solving other optimization problems such itinerary optimization in transport scheduling problem as discussed in [10,11]. The framework of hyper-heuristics is shown by Fig. 1.

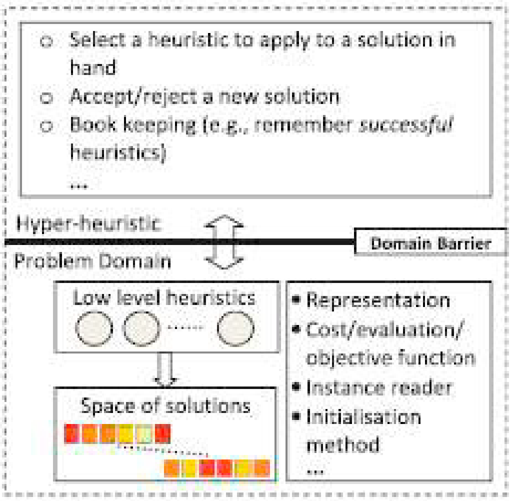


Fig. 1. The Hyper-heuristics Framework.

2.3. Tabu search

Tabu or *taboo* comes from Tongan, a Polynesian language, which has been used by aboriginal people in Tongan island which means things that cannot be touched because they are sacred. The idea of Tabu Search is appeared by Fred Glover in 1986 who agreed with the local search method to solve optimization problems [12]. Glover and Laguna define Tabu Search as one of the meta-heuristic methods that guides local search heuristic procedures to explore solutions space beyond local optimality. The problem that is feared will emerge is the search for solutions that are trapped in the local optimum area.

To prevent that problem, maintain a taboo list containing steps that meet some taboo limitation criteria. Previous steps included in the taboo list are prohibited from being included in certain iterations, usually called tabu tenure. Taboo tenor determines how long a step remains taboo. But the mechanism, called aspiration criterion, is used to replace the taboo status of a movement. The aspiration criterion is a better fitness value (increased cost function) where the taboo movement is changed to a non-taboo step when producing a better solution [13]. The set of candidates which are sets that contain elements that will later form a solution.

The Tabu Search algorithm has three main strategies including (1) forbidding strategy, (2) freeing strategy, and (3) short-term strategy. (1) Forbidding strategy is to control what parameters will enter the taboo list. (2) Freeing strategy is to control what and when it comes out of the taboo list. (3) Short-term strategy is to organize collaboration between the forbidding strategy and freeing strategy [14].

2.4. Variable neighborhood search

Variable Neighborhood Search (VNS) was introduced by Mladenovic and Hansen. This algorithm is based on a strategy using more than one environmental structure and changing the structure systematically during local searches. This helps VNS to explore environments that are far from current solutions and move to new solutions [15].

Variable Neighborhood Search consists of three stages: (1) Shaking, is scrambling existing solutions, (2) Local Search, is looking for new solutions in the solution area, and (3) Move, is the action for new solutions produced. If the new solution is better than before, it will replace the old solution [15,16]. Local search is applied repeatedly to get the optimum local from the current solution. Initially, the basic VNS approach was a derivative method that did not receive a worse solution to get out of local choices because the environmental structure varied regularly. Environmental structure can be carried out during the search, because local optimum in one environmental structure is not necessarily a local choice in other environmental structures [16].

2.5. Tabu - variable neighborhood search

The combining of the two algorithms is expected to produce a more optimal solution. The Tabu Search algorithm is used to handle environmental structures that do not lead to new solutions received. Taboo restrictions are applied where the environmental structure will become taboo if the value of the new solutions is better than the value of the old solution and the solution is rejected by the applicable acceptance criteria. In [1] and [17] the hybrid of tabu search and variable neighborhood algorithm was employed for solving vehicle routing problem with time windows, whereas in [18] the hybrid algorithm was employed for solving the median cycle problem. However, to best of our knowledge there have not been prior works investigating tabu – variable neighborhood search for solving timetabling problem. Moreover, the previous works was implemented within meta-heuristic approach, not within hyper-heuristics framework as discussed in this paper.

3. Problem formulation

The model was developed based on the real-word course timetabling problem from the department of information systems, Institut Teknologi Sepuluh Nopember. In concise, the course timetabling problem can be formulated as follow.

- The decision variable is given in Equation 1.

$$X_{i,t,r} \begin{cases} 1, if \;\; event\ i\ is\ scheduled\ within\ timeslot\ t, \\ \qquad and\ room\ r \\ \\ 0, else \end{cases} \tag{1}$$

In which $\mathrm{i} = \{1,2,3, \ldots .\mathrm{I}\}$ is the sequence number of event, $\mathrm{t} = \{1,2,3, \ldots, \mathrm{T}\}$ is the sequence of timeslots, and $\mathrm{r} = \{1,2,3, \ldots, \mathrm{R}\}$ is the sequence number of the available rooms.

- The hard constraints; the hard constraints consist of:
  i. Each event must be scheduled exactly once (Equation 2)
  ii. There are not two events with a common student scheduled in the same timeslots (Equation 3)

iii. The number of students does not exceed the room capacity (Equation 4)
iv. They are not two different events scheduled in the same toom (Equation 5)

$$\sum_{i=1}^{I}\sum_{t=1}^{T}\sum_{r=1}^{R} \boldsymbol{X_{itr}} = \mathbf{1} \tag{2}$$

$$\sum_{i=1}^{I-1}\sum_{j=i+1}^{I} C_{ij}.V_{ij} = 0 \tag{3}$$

$$V_{ij}\begin{cases} 1\ jika\ t_i = t_j \\ 0 \end{cases}$$

C_{ij} = the number of students enrolled in both event i and j

V_{ij} = vector whether event i and j are scheduled in the same timeslot

t_i = timeslot when event i is scheduled.

$$\sum_{i=1}^{I} S_i X_{it} \leq \sum_{r=1}^{R} P_r \tag{4}$$

S_i = the number of student enrolled in event i

P_r = the capacity of room r

$$\sum_{i=1}^{I}\sum_{t=1}^{T}\sum_{r=0}^{R} X_{itr} = 1 \tag{5}$$

$$x_{itr}\begin{cases} 1, \quad \text{if timeslot t,} \\ \text{and room r are scheduled for event i} \\ 0, \quad \text{else} \end{cases}$$

$$P_1 = \sum_{s=1}^{S}\sum_{h=1}^{H} X_{sh3} \tag{6}$$

S is the number of students, H is the number of days, X_{sh3} is 1 if the student s enrolled three event in day h.

$$P_2 = \sum_{s=1}^{S} \sum_{h=1}^{H} X_{sh1} \tag{7}$$

X_{sh1} is 1 if the student s enrolled only 1 event in day h.

$$P_3 = \sum_{s=1}^{S} \quad \sum_{t=1}^{T} X_{tmd3} \tag{8}$$

T is the total number of timeslots, X_{tmd3} is 1 if the student enrolled in event that is scheduled in timeslots that divisible by 3.

- The soft constraints; the soft constraints consist of:
  i. The students should not attend more than three events per day (Equation 6)
  ii. The students should not only attend one event per day (Equation 7)
  iii. The students should not attend the last event of the day (Equation 8).

- The objective function is to minimise the soft constraints, i.e. minimise P1+P2+P3.

4. Methodology

In general, the methodology used in this study could be divided into two main parts: the method to generate initial feasible solution and the method to optimize the initial solution. To generate initial solution, a greedy algorithm was employed whereas to optimize the initial solution, tabu-search algorithm was employed. The detail of these methods is explained below.

4.1. Greedy algorithm

Generating initial solution will be used as input to the optimization process is carried out by using the Greedy Algorithm. The initial solution contains the timeslot and the room allocated for each event. To implement this algorithm takes some initial information such as a list of conflicts for each event, the number of timeslots, the number of students per event and the capacity of each timeslot. The events of courses are sorted based on the number of available timeslot and the number of conflicts, before to be assigned to timeslots and room sequentially. Event with least number of available timeslot and highest total number of conflicts is assigned to timeslot and room first. Followed by the event as in the order, until all events are assigned to timeslot and room.

4.2. Tabu-variable neighborhood search

Using initial solution from the previous stage, in this stage the initial solution is perturbed in order to get more optimal solution. Tabu search algorithm is employed within VNS to select the best solution generated during optimization. By using the Tabu Search algorithm there will be a taboo list that can help in saving strategies that result in a higher penalty score, so that it will not be used first. Tabu search pseudocode combined with the VNS algorithm can be seen in Fig. 2.

```
Initiation :
    1.  Select neighborhood structure nk, k = {1,2,3,….,K}:
        for i=0 to size k
    2.  [string][]  sol ← string[][]
    3.  Set End
    4.  [string][]  bestSol ← string[][]
Repeat :
    1.  for i = 0 to size k
        a.  shaking (generate random solution sol from nk
        b.  local search : random.math
        c.  move or not
            if ((f(bestSol) < s) or (f(bestSol) is accepted by the
            acceptance criterion)) then
               sol ←  bestSol;
               set k ← 1;
               while k is in the tabulist and k < K
               k ← k+1;
               continue the search with nk;
            else
               insert k to the tabulist;
               set k ← k+1;
               increase the tabu length by 1;
               if tabu length > tabu tenure
                  release the first neighbourhood structure        from
            the tabu list;
               while k is in the tabulist and k < K
               k ← k+1;
```

Fig. 2. Tabu VNS Pseudocode.

5. Experimental results and discussion

The data, upon which the experiments were conducted consists of two datasets [19]. The first dataset is taken from the course timetabling problem in Odd semester, academic year 2017/2018. The second dataset is taken from course timetabling problem in Even semester, in the same academic year. Both datasets are real-world datasets that can be summarised by Table 1. Each course could have one or two events a week based on its credits. The timetabling problem is basically the problem of assigning each event into timeslots and room that satisfy all the hard constraints and minimize the soft constraints.

The initial solution was generated using sequential greedy algorithm. Within this algorithm, the event was sorted based on the number of conflicts with other events, i.e. events with common students. The event with highest number of conflicts was scheduled first, i.e. assigned to timeslots and room first. The experimental results showed that the proposed greedy algorithm could results in feasible solutions for both datasets. An example of initial feasible solution can be observed in [19].

Table 1. The Datasets.

No.	Name	Number of Student	Number of Courses	Number of Timeslots	Number of Rooms
1	Odd	651	32	15	11
2	Even	538	36	15	11

Initiated with the feasible solution generated with the greedy algorithm, the proposed hyper-heuristics, based on Tabu search and Variable Neighborhood Search aims at minimizing the soft constraints while maintaining the hard constraints. The proposed algorithm was run 11 times within 5000000 iterations per run (approximately less than 2 minutes). The average penalty from the experimental over the two datasets are summarized by Fig. 3.

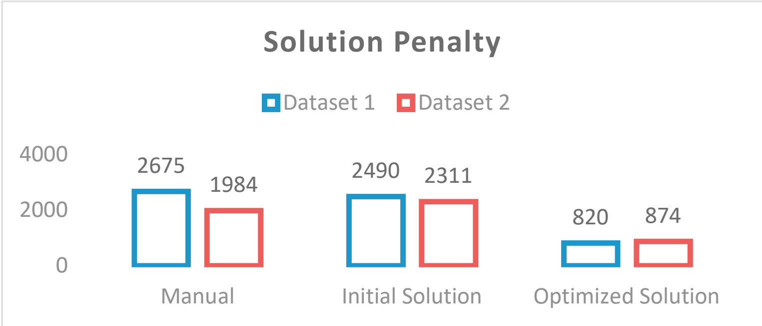
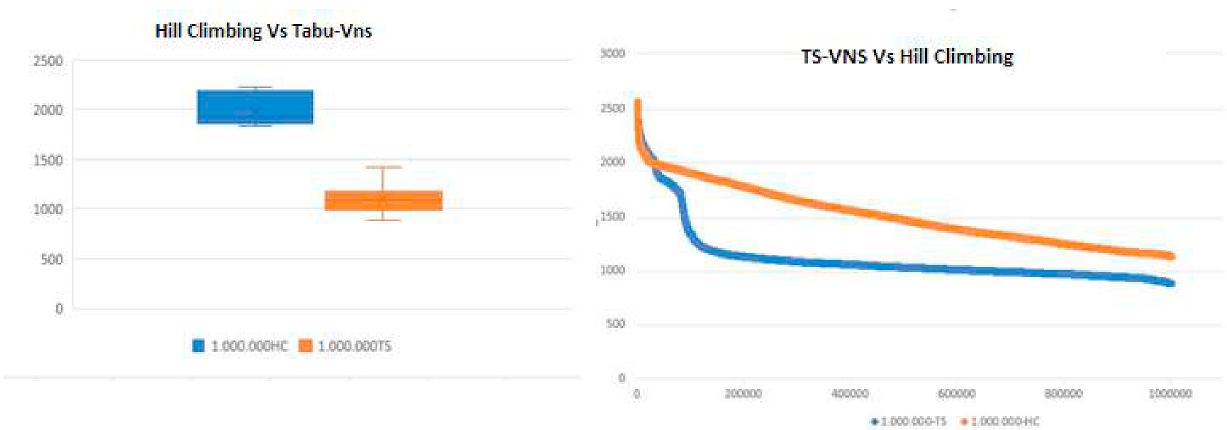


Fig. 3. The experimental results.

It is very clear from the figure that the solutions generated by the proposed algorithm are much better than the solution generated manually. To be more specific it could improve by 1855 and 1110 compared to the manual schedule. In real-world application it could be significantly less students should attend class more than three times a day. In addition, compared with benchmarking heuristic algorithm, i.e. hill climbing algorithm, the proposed algorithm significantly outperforms the hill climbing algorithm as shown by Fig. 4. Fig. 4 (a) compares the performance in terms of the solution quality measured by the penalty whereas Fig. 4 (b) compares the performance in terms of the algorithm trajectory search.



(a) (b)

Fig. 4. Hill Climbing Vs Tabu-VNS. (a) compares the performance in terms of the solution quality measured by the penalty whereas; (b) compares the performance in terms of the algorithm trajectory search.

6. Conclusion and future works

Based on the result of this research, it can be concluded that the proposed algorithm could automate the process of real-world course timetabling with the results significantly better than the timetable produced manually. In terms of the performance of algorithm, the proposed algorithm, tabu-search variable neighborhood search based hyper-heuristics outperforms the benchmarking heuristics algorithm, i.e. hill climbing algorithms. However, since it is the first work tested over the new provided datasets, there are still many aspects to be improved as future works. Further investigations on new strategies within the hyper-heuristic approaches could be proposed to improve the performance of the algorithm as well as further insight on the problem formulations.

References

[1] Belhaiza, S., P. Hansen, and G. Laporte. (2014) "A Hybrid Variable Neighborhood Tabu Search Heuristic for The Vehicle Routing Problem With Multiple Time Windows." *Comput. Oper. Res.* **52**: 269–281.

[2] Rochman, A. (2012) "Penjadwalan Kuliah Menggunakan Metode Constraints Programming Dan Simulated Annealing [Title in English: *Course Timetabling using Constraints Programming Method and Simulated Annealing*]", in *Seminar Nasional Aplikasi Teknologi Informasi* **2012**: 15–16.

[3] Rossi-Doria, O., and M. Sampels. (2003) "A Comparison of the Performance of Different Metaheuristics on the Timetabling Problem." *Pract. Theory Autom. Timetabling*. pp. 329–351.

[4] Qu, R., E. K. Burke, B. McCollum, L. T. G. Merlot, and S. Y. Lee. (2009) "A Survey of Search Methodologies and Automated System Development for Examination Timetabling." *J. Sched.* **12 (1)**: 55–89.

[5] Beligiannis, G.N., C. N. Moschopoulos, G. P. Kaperonis, and S. D. Likothanassis. (2008) "Applying Evolutionary Computation to The School Timetabling Problem: The Greek Case." *Comput. Oper. Res.* **35 (4)**: 1265–1280.

[6] Muklason, A., A. J. Parkes, E. Özcan, B. McCollum, and P. McMullan. (2017) "Fairness in Examination Timetabling: Student Preferences and Extended Formulations." *Appl. Soft Comput. J.* **55**.

[7] Muklason, A., P. C. Bwananesia, S. H. Y. T, N. D. Angresti, and V. A. Supoyo. (2018) "Automated Examination Timetabling Optimization Using Greedy-Late Acceptance-Hyperheuristic Algorithm", in *2018 International Conference on Electrical Engineering and Computer Science (ICECOS)*. pp. 201–206.

[8] Muklason, A. (2017) "Solver Penjadwal Ujian Otomatis Dengan Algoritma Maximal Clique dan Hyper-heuristics [Title in English: *Solver for Automatic Exam Scheduler with Maximal Clique Algorithm and Hyper-heuristics*]", in *Seminar Nasional Teknologi Informasi Komunikasi dan Industri*. pp. 94–101.

[9] Kalender, M., A. Kheiri, E. Özcan, and E. K. Burke. (2012) "A Greedy Gradient-Simulated Annealing Hyper-Heuristic for a Curriculum-Based Course Timetabling Problem," *2012 12th UK Work. Comput. Intell. UKCI 2012* **(1)**: 1–8, 2012.

[10] Wijaya, J. S., W. Anggraeni, A. Muklason, F. Mahananto, E. Riksakomara, and A. Djunaidy. (2017) "Advanced Traveler Information System: Itinerary Optimization as an Orienteering Problem Using Iterative Local Search-Hill Climbing Algorithm." *Procedia Comput. Sci.*, **124**: 173–180.

[11] Yoga, I. W. A. K., *et al.* (2018) "Advanced Traveler Information Systems: Itinerary Optimisation Using Orienteering Problem Model and Genetic Algorithm", in *2018 International Conference on Information Technology Systems and Innovation (ICITSI)*. pp. 454–459.

[12] Setemen, K. (2009) "Optimasi Generate Jadwal Mata Kuliah Menggunakan Algoritma Genetika dan Tabu Search [Title in English: *Optimize Course Schedule Timetabling Using Genetic Algorithm and Tabu Search*]", in *Seminar Internasional Revitalisasi Pendidikan Kejuruan dalam Pengembangan SDM Nasional*. pp. 783–791.

[13] Abdullah, S. (2006) "Heuristical Approaches for University Timetabling Problem." [master's thesis], Nottingham, The University of Nottingham.

[14] Pam, D. T., and D. Karaboga. (2000) *Intelligent Optimisation Techniques. Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks*.

[15] Hansen, P., N. Mladenović, and D. Perez-Britos. (2001) "Variable Neighborhood Decomposition Search." *J. Heuristics* **7 (4)**: 335–350.

[16] Abdullah, S., E. K. Burke, and B. Mccollum. (2005) "An Investigation of Variable Neighbourhood Search for University Course Timetabling", in *The 2nd Multidisciplinary International Conference On Scheduling: Theory and Applications (MISTA)*. pp. 413–427.

[17] Paraskevopoulos, D. C., P. P. Repoussis, C. D. Tarantilis, G. Ioannou, and G. P. Prastacos. (2008) "A Reactive Variable Neighborhood Tabu Search for The Heterogeneous Fleet Vehicle Routing Problem With Time Windows." *J. Heuristics* **14 (5)**: 425–455.

[18] Moreno-Pérez, J. A., J. Marcos Moreno-Vega, and I. Rodríguez Martín. (2003) "Variable Neighborhood Tabu Search and Its Application to The Median Cycle Problem." *Eur. J. Oper. Res.* **151 (2)**: 365–378.

[19] Muklason, A. (2019) "Course Timetabling Problem Open Dataset." Available from: http://intip.in/itscttdatasets. [Accessed 30th June 2019].