
A GRAPH-BASED SOLVER IN THE UNIVERSITY COURSE TIMETABLE SCHEDULING

INTERN REPORT

Chongfeng Ling
1716474

December 1, 2021

ABSTRACT

None

Keywords Timetable Scheduling · Graph Coloring · Tabu Search · Attention · Reinforcement Learning

Contents

1	Introduction	3
2	Literature Review	3
3	Problem Description	4
3.1	Mathematical formulation: A Graph Coloring representation	4
3.2	Mathematical programming to optimal solution	5
4	Research Approach	5
4.1	Methodology	5
4.1.1	Tabu Search	5
4.2	Data collection	6
5	Execution Plan	6

1 Introduction

Timetable scheduling is a practical problem with applications in several area including transportation, hospital, education and so on. Every semester an university will develop timetables for teach activities and examination to meeting students and staffs' requirements and school hardware source limitation. Credit to pervious works, university timetabling has be divide into two interrelated subproblem: timetabling subproblem and grouping subproblem (Werra & Hertz 1989). While a lot of studies has been spent on this topic, there is still a big gap between algorithm result and practical timetable (McCollum 2006) mainly due to various constraints in different university and large scale of data. Consider these constraints with a large volume of data, timetable scheduling is a NP-hard or NP-complete problem (Even et al. 1975), which means it can not be solved in a polynomial time with large scale problem data. Hence, up to now most of solver is based on heuristic algorithm and its variant like meta-heuristic and hyper-heuristic. This paper attempts to implement some heuristic algorithm and introduce a new way called Attention Mechanism with Reinforcement Learning to solve course timetable problem in XJTU.

The structure of this paper is organized as follow. Section 2 is a review of the development in timetable scheduling including algorithms, computer system and benchmarks. Section 3 is a problem description based on graph theory. In section 4 we present methodologies including Tabu Search algorithm and attention with reinforcement learning. Specific execution plan with timeline is stated in Section 5.

2 Literature Review

Owing to the enormous achievements in university timetable problem research, numerous literatures has been published and areas of concern to them are including model representing, meta-heuristic algorithm, computer system and benchmarks.

In 1985, Werra (1985) stated a formal way to model course-teacher timetabling and provided formulations in both graph edge coloring and graph node coloring. Then Hertz & Werra (1987) solved big random graph coloring problem by tabu algorithm and the TABUCOL procedure, compared with annealing algorithm, the CPU-time was much shorter and furthermore, for some unsolved graph, tabu algorithm could indicate the "bad" edges that needed to be reduced. The principles and illustrations of Tabu Algorithm were given by Werra & Hertz (1989) and later Glover (1990) mentioned that the advantage of Tabu Search compared other meta-heuristic algorithm owing to its long-term memory. Hertz (1991) in his paper used Tabu Algorithm to solved timetabling problem and due to hard constraints can not guarantee the existence of a feasible solution, he first defined the feasible solution respect to soft constraints and optimized it by hard constraints. In addition, he mentioned another problem that group students into class which they have the same courses in the same class. Costa (1994) made a detailed description about timetable problem and mathematical formulation. Based on the property of meta-heuristic, he generated a general Tabu Algorithm which can be adapted under various constraints and used in different university and colleges. In 1997, Werra (1997) added a new constraint to spread lectures uniformly across a set of periods and proved some existence of solutions under some typical constraints. Schaerf (1999) made a survey about how heuristic algorithm could assign timetable automatically. Carter (2000) described a comprehensive university timetable system in Waterloo including system structure and algorithm phase. Additionally, he introduced decomposition in both student section and timetable which will reduce algorithm complexity and conflicts. With the development of enormous heuristic algorithms, Silver (2004) made a review of them and stated two ways to evaluate their performance. Gendreau & Potvin (2005) took another combinatorial optimization problem (OSP) TSP as an examples and summarized usage of different meta-heuristic in one OSP. A hyper-heuristic algorithm based on Tabu Algorithm was used to solved examination by Hussin (2005) and Kendall & Hussin (2005) which is parameter free. McCollum (2006) gave a overview on gaps of timetabling problem between theory and practice and bridged the gaps between the two. Moreover, The Second International Timetable Competition of 2007 also aimed to reduce the gaps and divided problem into three tracks, though in a general system the last two problems are not isolated with each other (*UniTime | University Timetabling* n.d.). To reduce the difficult of finding a feasible initial solution, Burke et al. (2007) create heuristic algorithm based on graph degree and operate algorithm in a hyper-heuristic search space. Similar to Hertz & Werra (1987), Tuga et al. (2007) generated a hybrid-heuristic algorithm and change two kinds of constraints to get a feasible solution in a diversification space. Burke et al. (2012) used a formulation to reduce the number of soft constraints to bounded the complexity of problem. Kristiansen & Stidsen (2013) and Johnes (2015) made a review of timetabling scheduling and student section. In addition, Kristiansen & Stidsen (2013) stated most of previous practical research in timetable were founded that they were based on simulation dataset. Hence they introduced a open-source dataset in Denmark university and its format description. Though tabu Search with its variants made a great deal in timetabling problem, Fazel Zarandi et al. (2020) affirmation the efficiency of hyper-heuristic algorithms but more attention was needed to identify which method dominated the search space of optimization problems. Due to the complexity and urgent demands, there

are some commercial and open-source tools published in the past several decades including Oracle Peoplesoft and *UniTime* | *University Timetabling* (n.d.). With the development of Neural Network, Fazel Zarandi et al. (2020) stated the advantages and disadvantages of some neural network algorithm and Tabu Search employed in scheduling. While attention mechanism successfully employed in some NLP models with transformer framework (Ashish Vaswani et al. 2017, Devlin et al. 2019), Kool et al. (2019) had used attention mechanism with reinforcement learning to solve TSP problem with up to 100 nodes.

3 Problem Description

In this section, we follow the terminology and problem descriptions by Werra (1985). According to the Wren (1996), timetabling problem is defined as the allocation to arrange resources into space and time subjects to constraints such that satisfies a set of desirable objectives as many as possible. We will firstly definite sources in university and then list constraints under XJTLU requirement. One curriculum contains several courses while each course could be repeated more than once and thus split into multiple sections. The problem of finding the best grouping of students into corresponding course section is called grouping problem. Normally in every week several lectures with corresponding teachers are hold respect to the course. The comprehensive definitions of resources that involved in XJTLU are listed as follow:

- Teacher set $T = \{t_1, \dots, t_j\}$
- Class set $C = \{c_1, \dots, c_i\}$. A class is a group of students who have the same curriculum.
- Classroom set $CR = \{cr_1, \dots, cr_{ncr}\}$
- Requirement matrix $R = (r_{ij})$ gives the number of lectures involving c_i and t_j during one week or day.
- A period is a day corresponding to weekly scheduling and each timeslot is a period in daily scheduling.
- Course set $CO = \{co_1, \dots, co_{nco}\}$. A course is defined by
 1. a set of teachers
 2. a set of classes
 3. a set of lectures $L = \{\ell_1, \dots, \ell_{n1}\}$
 4. a set of course sections.

Based on the above definitions of sources, constraints of the timetable problem are as follow:

1. teacher overlaps: a teacher cannot be involved simultaneously in more than one lecture.
2. class overlaps: a class cannot be involved simultaneously in more than one lecture.
3. classroom overlaps: a classroom cannot be involved simultaneously in more than one lecture.
4. period constraints: the duration of lectures could be one or two hours.
5. pre-assignment constraints: the lectures are preassigned to a set of specific periods or classrooms.
6. teacher unavailability: a lecture involving a teacher t_j cannot be scheduled at a period during which t_j is not available, including lunch break and university free afternoon (i.e. Wednesday afternoon in XJTLU)
7. geographical constraints: Two lectures given in two distant classroom should be scheduled consecutively if and only if there is sufficient time for moving one classroom to another.
8. compactness constraints: each teacher and student wants a schedule with a minimal number of holes and isolated lectures.
9. distribution constraints: the identical lectures (i.e. the lectures of a same course) should be spread as uniformly as possible in weekly scheduling.

Depend on university management, these constraints should be spilt into two parts: one is hard constraints and the other is soft constraints. While optimizing, a feasible solution is the one which satisfied all hard constraints, moreover, the optimal one is a feasible solution and satisfies all soft constraints.

3.1 Mathematical formulation: A Graph Coloring representation

Consider a basic course scheduling model in daily scheduling. For one course co_a with a set of lectures $L = \{\ell_1, \dots, \ell_b\}$, we denote a lecture-node m_{ab} for each course and lecture. Due to the class overlaps constraint, all

pairs of lecture note in course co_a are connected by edges. While assume all course have no sections, if there is a student taking both courses co_{a1} and co_{a2} , we introduce an edge between every pair of lecture node m_{a1b} and m_{a2b} . The feasible course scheduling among p periods is respect to the node coloring of graph G with p colors. An example is given in Figure 1. Here we have 3 courses and each of them has 1, 2 or 3 lectures. Student group A takes courses co_1 and co_2 , another group B takes co_2 and co_3 . A feasible solution is drawn by 5 colors, which means at least 5 periods is needed to assign courses without conflicts.

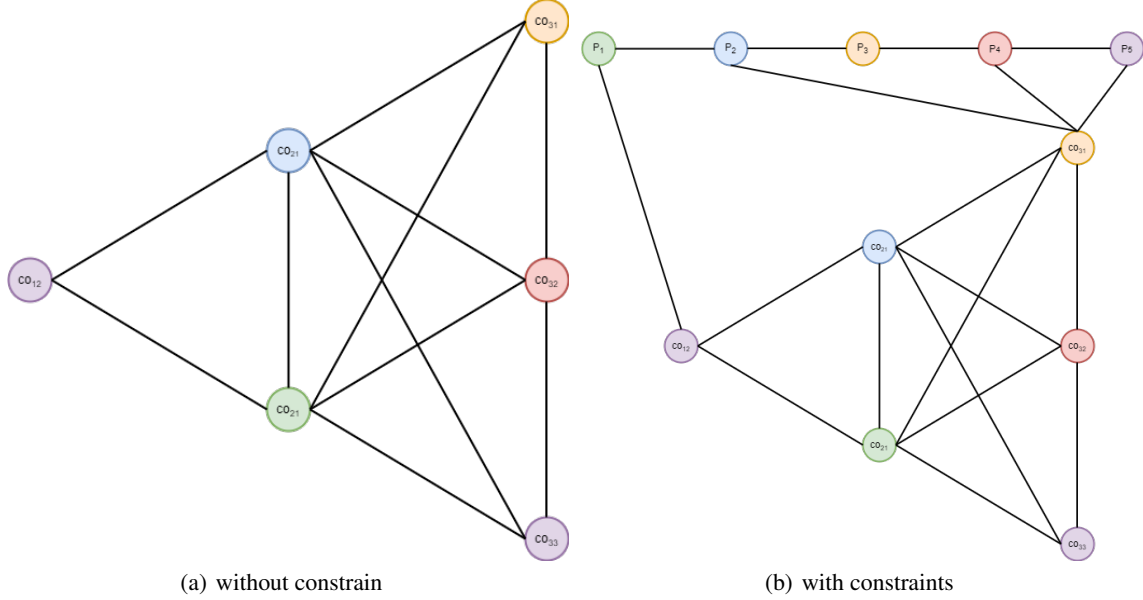


Figure 1: Graph Coloring representation

With pre-assignment constraints and teacher unavailability, we introduce two constraints samples: (1). co_1 not scheduled at period 1; (2). one lectures of K_3 at period 1 or 3. A set of period nodes are added to Figure 1(a) and then we get Figure 1(b). Easy to prove that pre-assignment of periods are equal to teacher time unavailability.

3.2 Mathematical programming to optimal solution

While feasible solution is exists, or we are supposed to evaluate one from some bad solutions where have conflicts in hard constraints, we introduce an objective function $\sum_{i=1}^q \sum_{k=1}^p C_{ik} y_{ik}$, $y_{ik} = 1$ if a lecture of course co_i is scheduled at period k and $y_{ik} = 0$ otherwise; C_{ik} is the cost function to indicate the plausibility of assignments. Hence, the problem is to find the best global optimal solution under constraints.

4 Research Approach

4.1 Methodology

4.1.1 Tabu Search

Tabu Search is a heuristic algorithm designed for finding a global optimal points and has been used for solving combinatorial optimization problem including graph node coloring, large scale timetabling and TSP efficiently. The basic process is given an feasible solution as initial points and an aspiration function to evaluate results, moving the initial point to another solution in the neighbor of initial point and making comparison and recording the better solution with aspiration function. The algorithm does not stop until get a global optimum or reach the max number of iterations. The core of Tabu Search algorithm is tabu list T that all moves back to current point are forbidden in the next $|T|$ iterations.

The general Tabu Search technique is shown as follow:

Initialization

```

 $s :=$  initial solution in  $X$ 
 $nbiter := 0$ 
    (* current iteration *)
 $bestiter := 0$ 
    (* iteration when the best solution has been found *)
 $bestsol := s$ 
    (* best solution *)
 $T := \emptyset$ 
    initialize the aspiration function  $A$ ;
while ( $f(s) > f^*$ ) and ( $nbiter - bestiter < nbmax$ ) do
     $nbiter := nbiter + 1$ ;
    generate a set  $V^*$  of solutions  $s_i$  in  $N(s)$  which are either not tabu or such that
         $A(f(s)) \geq f(s_i)$ ;
    choose a solution  $s^*$  minimizing  $f$  over  $V^*$ 
    update the aspiration function  $A$  and the tabu list  $T$ ;
    if  $f(s^*) < f(bestsol)$  then
         $bestsol := s^*$ ;  $bestiter := nbiter$ ,
     $s := s^*$ ;

```

4.2 Data collection

There are two ways to collect dataset. The first is open-source datasets in UniTime collection (). The corresponding benchmark is also given. To deal real problem in XJTLU, we first try year 4 applied mathematics students and teachers with classrooms in Mathematics Building. Consider a large number of course sections, deal with the timetable of year 1 students is the third steps.

5 Execution Plan

- Programming system
 1. Data collection
 - (a) Online dataset: Reader function of XML file in Python. Timeline: Dec. 3 - Dec. 5
 - (b) XJTLU dataset:
- Algorithm
 1. Tabu Algorithm
 - (a) random graph coloring. Timeline: Dec. 2 - Dec. 4
 - (b) timetabling subproblem with online dataset. Timeline: Dec. 4 - Dec. 6
 - (c) grouping subproblem with online dataset. Timeline: Dec. 6 - Dec. 15
 - (d) real timetabling subproblem in XJTLU, Y4, AM. Timeline: Dec. 20 - Jun. 14
 - (e) real grouping subproblem in XJTLU, Y1. Timeline: Dec. 20 - Jun. 27
 2. graph decomposition algorithm. Timeline: Jun. 17 - Jun. 23
 3. Attention mechanism Reinforcement Learning
 - (a) Attention mechanism. Timeline: Jun. 24 - Jun. 30
 - (b) Reinforcement Learning. Timeline: Feb. 1 - Feb. 13
 - (c) Attention mechanism with RL in TSP: Feb.14 -Feb. 20
 - (d) real timetable scheduling problem in XJTLU.
- Skills
 1. Plot graph in Python.

References

- Ashish Vaswani, Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. & Polosukhin, I. (2017), 'Attention Is All You Need', *arXiv:1706.03762 [cs]*.
- Burke, E. K., McCollum, B., Meisels, A., Petrovic, S. & Qu, R. (2007), 'A graph-based hyper-heuristic for educational timetabling problems', *European Journal of Operational Research* **176**(1), 177–192.
- Burke, E., Marecek, J., Parkes, A. & Rudová, H. (2012), 'A branch-and-cut procedure for the Udine Course Timetabling problem', *Annals of Operations Research* **194**, 71–87.
- Carter, M. (2000), 'A comprehensive study of educational timetabling - a survey'.
- Costa, D. (1994), 'A tabu search algorithm for computing an operational timetable', *European Journal of Operational Research* **76**(1), 98–110.
- Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. (2019), 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding', *arXiv:1810.04805 [cs]*.
- Even, S., Itai, A. & Shamir, A. (1975), On the complexity of time table and multi-commodity flow problems, in '16th Annual Symposium on Foundations of Computer Science (Sfcs 1975)', IEEE, USA, pp. 184–193.
- Fazel Zarandi, M. H., Sadat Asl, A. A., Sotudian, S. & Castillo, O. (2020), 'A state of the art review of intelligent scheduling', *Artificial Intelligence Review* **53**(1), 501–593.
- Gendreau, M. & Potvin, J.-Y. (2005), 'Metaheuristics in Combinatorial Optimization', *Annals of Operations Research* p. 25.
- Glover, F. (1990), 'Tabu Search: A Tutorial', *Interfaces* **20**(4), 74–94.
- Hertz, A. (1991), 'Tabu search for large scale timetabling problems', *European Journal of Operational Research* **54**(1), 39–47.
- Hertz, A. & Werra, D. (1987), 'Using tabu search techniques for graph coloring', *Computing* **39**(4), 345–351.
- Hussin, N. M. (2005), 'TABU SEARCH BASED HYPER-HEURISTIC APPROACHES TO EXAMINATION TIMETABLING', p. 230.
- Johnes, J. (2015), 'Operational Research in education', *European Journal of Operational Research* **243**(3), 683–696.
- Kendall, G. & Hussin, N. M. (2005), An Investigation of a Tabu-Search-Based Hyper-Heuristic for Examination Timetabling, in G. Kendall, E. K. Burke, S. Petrovic & M. Gendreau, eds, 'Multidisciplinary Scheduling: Theory and Applications', Springer-Verlag, New York, pp. 309–328.
- Kool, W., van Hoof, H. & Welling, M. (2019), 'Attention, Learn to Solve Routing Problems!', *arXiv:1803.08475 [cs, stat]*.
- Kristiansen, S. & Stidsen, T. J. R. (2013), 'A Comprehensive Study of Educational Timetabling - a Survey'.
- McCollum, B. (2006), 'A Perspective on Bridging the Gap Between Theory and Practice in University Timetabling'.
- Schaerf, A. (1999), 'A Survey of Automated Timetabling', *Artificial Intelligence Review* p. 41.
- Silver, E. A. (2004), 'An overview of heuristic solution methods', *Journal of the Operational Research Society* **55**(9), 936–956.
- Tuga, M., Berretta, R. & Mendes, A. (2007), 'A Hybrid Simulated Annealing with Kempe Chain Neighborhood for the University Timetabling Problem', *6th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2007)* pp. 400–405.
- UniTime | University Timetabling (n.d.), <https://www.unitime.org/index.php?tab=0>.
- Werra, D. (1985), 'An introduction to timetabling', *European Journal of Operational Research* **19**, 151–162.
- Werra, D. (1997), 'The combinatorics of timetabling', *European Journal of Operational Research* **96**(3), 504–513.
- Werra, D. & Hertz, A. (1989), 'Tabu search techniques: A tutorial and an application to neural networks', *Operations-Research-Spektrum* **11**(3), 131–141.
- Wren, A. (1996), Scheduling, timetabling and rostering — A special relationship?, in G. Goos, J. Hartmanis, J. Leeuwen, E. Burke & P. Ross, eds, 'Practice and Theory of Automated Timetabling', Vol. 1153, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 46–75.