# A graph-based solver in the university course timetable scheduling
# 一个基于图着色的大学课程排课求解器

Interim Report

**Chongfeng Ling**
1716474
Department of Applied Mathematics
Xi'an Jiaotong-Liverpool University
Chongfeng.Ling17@student.xjtlu.edu.cn

**M.B.N. (Thijs) Kouwenhoven**
Supervisor
Head, Department of Physics
Xi'an Jiaotong-Liverpool University
t.kouwenhoven@xjtlu.edu.cn

April 15, 2022

## Contents

## Abstract

Timetabling problem is popular in various areas and can be formulated to graph coloring problem. We consider a simple kind - the university timetabling problem in XJTLU as our target problem. Due to the large scale number of people and constrains, we use and modify a heuristic algorithm as our solver to built up a timetable system. With the development of Neural Networks and its successful applications in other combinatorial problems, we also attempt to apply Reinforcement Learning to enhance our solver.


***Keywords*** Timetable Scheduling · Graph Coloring · Heuristic Algorithm · Tabu Search · Reinforcement Learning

# 1 Introduction

Timetable scheduling is a practical problem with applications in several areas including transportation, hospital and education. Every semester one university is supposed to 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 been divide into two interrelated subproblem: timetabling subproblem and grouping subproblem based on graph coloring(Hertz 1991). 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 of 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. Up to now most of solver is based on heuristic algorithm and its variants like meta-heuristic and hyper-heuristic. This paper attempts to implement some heuristic algorithm while introducing a new method Reinforcement Learning to built a system to solve course timetabling problem in Xi'an Jiaotong-Liverpool University (XJTLU).

The structure of this paper is organized as follow. Section 2 is a review of the development of 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 one improved version called TABCOL. Specific execution plan with timeline and challenges is stated in Section 5.

## 2 Literature Review

Since the 1980s, the connection between timetabling problem and graph coloring with heuristic algorithms has been established. 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 a large scale random graph coloring problem by tabu algorithm and the TABUCOL procedure that reduces number color from maximum value, 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) and Tuga et al. (2007) used Tabu Algorithm to solved timetabling problem and due to hard constraints can not guarantee the existence of a feasible solution, they first defined the feasible solution respect to soft constraints and optimized it by hard constraints. In addition, 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. To reduce the influence of parameter choice, a hyper-heuristic algorithm based on Tabu Algorithm was used to solved examination by Hussin (2005) and Kendall & Hussin (2005) which is parameter free. Galinier & Hertz (2006) modified TABCOL algorithm and made a comparison between four graph coloring method with different search space and search strategies. While an initial feasible solution is needed for all heuristic algorithm, Burke et al. (2007) create heuristic algorithm based on graph degree and operate algorithm in a hyper-heuristic search space reduce the difficult of finding a feasible initial solution.

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.

There are also some appreciated framework of timetabling systems. 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. In general, practical problems are more complexity than algorithm theory. McCollum (2006) gave a overview on gaps of timetabling problem between theory and practice and bridged the gaps between the two. 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. ALTUMA (Tesfaldet 2008) was the solver in the University of Asmara which using memetic algorithm, its performance was experienced and evaluated by read data in the university successfully. Moreover, UniTime (Müller & Rudová 2016) was an open-source system and successfully implemented in a large scale university.

## 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 constrains 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. In gegeral, one curriculum contains several courses while each course could be repeated more than once and thus split into multiple sections. The subproblem 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 $T = \{t_1, \ldots, t_j\}$
- Class set $C = (c_1, \ldots, c_i\}$. A class is a group of students who have the same curriculum.
- Classroom set $CR = \{cr_1, ..., cr_{ncr}\}$
- 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 $CO = \{co_1, ..., co_{nco}\}$. A course is defined by
  1. a set of teachers
  2. a set of classes
  3. a set of lectures $L = \{\ell_1, \ldots, \ell_{nl}\}$
  4. a set of course sections.

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

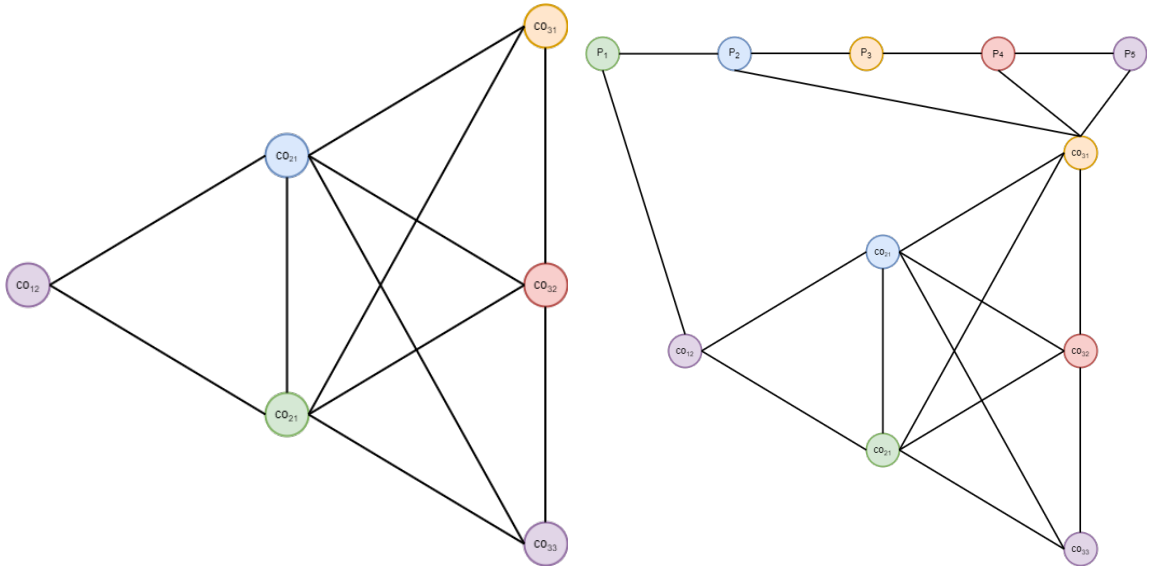
1. teacher overlaps: a teacher cannot be involved simultaneously in more than one lecture.
2. class overlaps: a class cannot be involved simultaneously in more than one lecture.
3. classroom overlaps: a classroom cannot be involved simultaneously in more than one lecture.
4. period constraints: the duration 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 $L = \{\ell_1, \ldots, \ell_{nl}\}$, we denote a lecture-node $m_{ab}$ for each course $co_a$ and lecture $l_b$. Due to the class overlaps constraint, all pairs of lecture note in course $co_a$ are connected by edges. While assume all courses 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.

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 see that pre-assignment of periods are equal to teacher time unavailability.




(a) without constrain (b) with constraints

Figure 1: Graph Coloring representation

# 4 Research Approach

## 4.1 Methodology

### 4.1.1 Tabu Search

Tabu Search is a heuristic algorithm designed for finding a global optimal point and has been used for solving combinatorial optimization problem including graph node coloring, large scale timetabling and TSP efficiently. The basic process is given a feasible solution as initial point and an aspiration function to evaluate results, moving the initial point to another solution in the neighbor of the initial point and making comparison and recording the better solution by 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. TABCOL(Hertz & Werra 1987) is a modified version of Tabu search for coloring of graph. It first initializes a random solution with a large enough number $k$, then use Tabu to reduce the conflicts of edges and obtain a feasible $k$-color graph as an initialization solution for next iteration until get a smallest color set for the graph.

**Initialization**
$s :=$ initial solution in $X$
nbiter $:= 0$
$(*$ current iteration $^*)$
bestiter $:= 0$
$(*$ iteration when the best solution has been found $^*)$
bestsol $:= s$
$(*$ best solution $^*)$
$T := \varnothing$
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,
$s := s^*$;

Figure 2: General Tabu Search Algorithm (Hertz 1991)

## 4.2 Data collection

There are two ways to collect data. The first is using open-source datasets in UniTime and also the corresponding benchmarks. To deal the real problem in XJTLU, we first collect high level students in applied mathematics, then consider course sections, data of students with fundamental mathematics couse is the third benchmark for our system.

# 5 Execution Plan

## 5.1 Timeline

- Programming system
  1. Data collection. Timeline: before the winter holidays.
     (a) Collect and save online data in University Course Timetabling Data Format (v2.4) [1], write and save data in Python.
     (b) Consider the course section and complexity of problems, we collect real data in XJTLU, 2021-2022 academic year, as follow:
        i. Year 4 applied mathematics students
        ii. Year 4 applied mathematics and finical mathematics students
        iii. Year 1 students with fundamental mathematics course.
- Algorithm
  1. Tabu Algorithm improvement
     (a) general TABCOL implemented in timetabling subproblem and grouping subproblem. Timeline: Dec. 1 - Dec. 10.
     (b) use matrix to speed up Tabu Search. Timeline: Dec. 10 - Dec. 20.
     (c) try hybrid method to find the feasible initial solution for Tabu search automatically. Timeline: Dec. 20 - Jun. 10.
     (d) use graph decomposition algorithm to reduce edges in the graph. Timeline: Jun. 10 - Jun. 25.
  2. Attention Mechanism and Reinforcement Learning
     (a) Attention Mechanism. Timeline: Jun. 25 - Feb. 5
     (b) Reinforcement Learning. Timeline: Feb. 5 - Feb. 20
     (c) Attention Mechanism with RL in TSP. Timeline: Feb. 20 - Feb. 28
     (d) real timetable scheduling problem in XJTLU. Timeline: semester 2.

## 5.2 Skills and Knowledge

1. **Graph theory**: learn how to color a graph and the method to reduce edge and decomposition of a big graph.
2. **Tabu Search**: learn its idea and variants in graph coloring and scheduling area.
3. **Reinforcement Learning**: it is a fast developed Neural Network and more applications appears in combinatorial operational problems. Study its principle and usage in scheduling problem.
4. **Plotting graph in Python**: there are many packages to plot in python, learn one of them to obtain visual, or even interactive graphs to help understand.

## 5.3 Challenges

1. **Lack of computing resources**: Heuristic algorithm is an efficient way to solver graph coloring problem and scheduling problems, due to the complexity of this class of NP-hard problems, however, as nodes of the graph increasing, using Tabu Search to get a result is also time-consuming. Taking TABCOL algorithm for coloring of graph [2] as an example, we create a series of graph which the probability of one edge in two random vertices is 0.5 and run it on two-core CPU machine. Figure 3 tells that with the increasing of vertices, running time grows exponentially.
2. **Deployment of RL algorithm**: Though RL algorithm has been developed in various areas for years, its applications in scheduling problems are rare. Luckily, we can learn another TSP to copy the experiences of how to use Reinforcement Learning algorithm in combinatorial problems, but still there is a long way to applied RL in our project and the difficulty can not be ignored.

[1] `https://www.unitime.org/uct_dataformat_v24.php`
[2] `https://github.com/pchervi/Graph-Coloring`

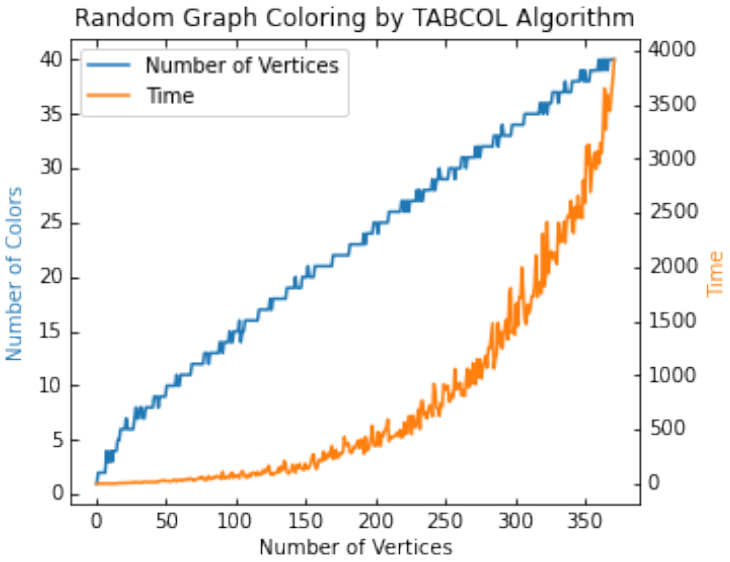



Figure 3: Random Graph Coloring by TABCOL Algorithm

## 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.

Carter, M. (2000), 'A Comprehensive Course Timetabling and Student Scheduling System at the University of Waterloo', *PATAT* .

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.

Galinier, P. & Hertz, A. (2006), 'A survey of local search methods for graph coloring', *Computers & Operations Research* **33**(9), 2547–2562.

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-Heristic Approachies to Examination Timetabling, PhD thesis.

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', *PATAT* .

Müller, T. & Rudová, H. (2016), 'Real-life curriculum-based timetabling with elective courses and course sections', *Annals of Operations Research* **239**(1), 153–170.

Schaerf, A. (1999), 'A Survey of Automated Timetabling', *Artificial Intelligence Review* p. 41.

Tesfaldet, B. T. (2008), 'Automated lecture timetabling using a memetic algorithm', *ASIA-PACIFIC JOURNAL OF OPERATIONAL RESEARCH* **25**(4), 451–475.

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.

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.