**Final Report**

July 29, 2024

CS/ECE/ISyE 524 — Introduction to Optimization — Summer 2024**University Course Timetabling Problem**

Evelyn Yeh ([eyeh6@wisc.edu](mailto:eyeh6@wisc.edu)), Yi-Chun Chen ([chen2524@wisc.edu](mailto:chen2524@wisc.edu))

**Table of Contents**

1. Introduction

2. Mathematical Model

3. Solution

4. Results and Discussion

5. Conclusion

6. References

1. **Introduction**

Timetabling problems are a specific type of scheduling problem that manages the assignment of events into timeslots. These assignments must adhere to numerous hard and soft constraints. The goal is to find a valid assignment that satisfies all hard constraints while maximizing the satisfaction of soft constraints . [1]

Timetabling problems have many variations and applications across different fields, such as airport slot scheduling, sports scheduling, logistics scheduling, and university timetabling. In this project, we focus on the University Course Timetabling Problem (UCTP), specifically the allocation of instructors to courses within fixed timeslots over a single week. [2]

Our scenario includes several classes, each with one or more courses that need a fixed number of timeslots each week, with a single corresponding instructor for each course. Each weekday has a fixed number of timeslots available for all the classes.

In addition to ensuring the schedule has no conflicts, we consider two types of hard constraints: instructor availability and specific timeslot requirements for certain courses. The soft constraint we focus on is minimizing the number of repeated courses in a single weekday, as this is generally disliked by both instructors and students.

**1.1 Handcrafted Data**

To confirm our model design works as intended, we start with simple handcrafted data with a small number of classes and instructors. We use two classes, nine instructors, and seven different courses, with four timeslots per weekday.

The following table shows the handcrafted course data created for the proof of concept:

|  |  |  |  |
| --- | --- | --- | --- |
| **Instructor** | **Course** | **# of timeslot for Class 1** | **# of timeslot for Class 2** |
| Mr. Cheese | English | 1 | 1 |
| Mrs. Insulin | Biology | 3 | 3 |
| Mr. Map | History-Geography | 2 | 2 |
| Mr. Effofecks | Mathematics | 0 | 4 |
| Mrs. Derivative | Mathematics | 4 | 0 |
| Mrs. Electron | Physics | 3 | 3 |
| Mr. Wise | Philosophy | 1 | 1 |
| Mr. Muscle | Sport | 1 | 0 |
| Mrs. Biceps | Sport | 0 | 1 |

Some instructors are unavailable for certain timeslots, and some courses must occur at specific times:

Unavailable Times:

|  |  |  |
| --- | --- | --- |
| **Instructor** | **Weekday** | **timeslot** |
| Mr. Effofecks | Monday | 1 |
| Mr. Effofecks | Monday | 2 |
| Mrs. Insulin | Wednesday | 1 |
| Mrs. Insulin | Wednesday | 2 |
| Mrs. Insulin | Wednesday | 3 |
| Mrs. Insulin | Wednesday | 4 |

Fixed Course Times:

|  |  |  |
| --- | --- | --- |
| **Course** | **Weekday** | **timeslot** |
| Sports | Thursday | 3 |

The code we used to represent this handcrafted data is as follows:

一張含有 文字, 螢幕擷取畫面, 數字, 軟體 的圖片

自動產生的描述

1. **Mathematical Model**

To find a valid schedule, we need to ensure that each class has all the required courses allocated with no extras. This is done by adding constraints for all courses of a certain class, ensuring the total number scheduled equals the number planned for the week.

**Constraints**

* Each instructor can be scheduled for at most one course per timeslot.
* Each class must have all required courses assigned without unnecessary courses.
* No instructor or class can have multiple courses scheduled in a single timeslot.
* Upper and lower bounds are set for each course for every timeslot to implement additional hard requirements (unavailable times and fixed course times).

**Soft Requirement**

To minimize repeated courses on a single weekday, we calculate the total count of the same course over a single day, subtract 1, and clamp the value to at least 0. This way, having a single course in a day is not penalized, and having no course in a day is not rewarded. The sum of this value across all weekdays is the penalty score for a schedule, which we aim to minimize.

* + 1. **UCTP Model**

**Data**

* *N* – Set of timeslots per day
* *M* – Set of weekdays
* *K* – Set of instructors-course pairs
* *B* – Set of classes
* *ckb* – Number of courses by instructor for class
* *uijkb* – Upper bound of scheduled courses by instructor at timeslot of weekday for class
* *lijkb* – Lower bound of scheduled courses by instructor at timeslot of weekday for class

**Variables**

* *xijkb* – Whether course by instructor for class is scheduled for timeslot of weekday

**Model**一張含有 文字, 字型, 螢幕擷取畫面, 筆跡 的圖片

自動產生的描述Model Constraints:

(1) The number of courses scheduled for every instructor at each timeslot must be at most 1.

(2) All required courses for each class must be assigned, and no unnecessary courses are scheduled.

(3) and (4) No conflicts occur for both instructors and classes.

(5) and (6) Upper and lower bounds are implemented to enforce additional hard requirements.

1. **Solution**

We transform the unavailable instructor and fixed course data into suitable data structures. 一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

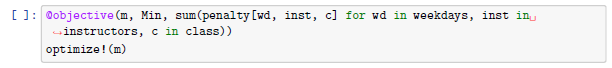
自動產生的描述We then set up the model and add the constraints and objectives.

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

For the objective function, we address the complication of clamping the penalty of repeated courses to greater or equal to zero. We use an auxiliary binary variable to formulate a constraint for the penalty function, ensuring any negative value is set to zero . [3]  
一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述  
We can now add the objective to the model and run the optimizer.



1. **Results and Discussion**

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 數字, 字型 的圖片

自動產生的描述

The optimizer finishes after 34 iterations. We output the generated schedule using PrettyTables. Observing the schedule, we confirm our model minimizes repeated courses while maintaining all hard requirements.

Our model supports slightly more complex scenarios than initially designed. Different classes can have different available timeslots, achieved by setting the upper bound of unavailable timeslots to zero. However, the model has limitations, such as treating instructors assigned to multiple courses as separate pairs, leading to potential conflicts not being identified.

1. **Conclusion**

The implemented UCTP model effectively schedules courses while adhering to hard constraints and minimizing repeated courses. Further refinements could address identified limitations and support more complex scenarios.

1. **References**

[1] Tomáš Müller, “*Constraint-based Timetabling”*, Ph.D. Thesis, Faculty of Mathematics and Physics, Charles University in Prague, Prague, 2005.

[2] Nancy Maribel Arratia-Martinez, Cristina Maya-Padron, and Paulina A. Avila-Torres, “University Course Timetabling Problem with Professor Assignment” Mathematical Problems in Engineering. [Online]. Available: <https://onlinelibrary.wiley.com/doi/10.1155/2021/6617177>

[3] user477602, linear programming set a variable the max between two another variables, URL (version: 2017-09-27): <https://math.stackexchange.com/q/2447498>