

Towards A Unified Timetabling Model

Jeffrey H. Kingston

The University of Sydney

Introduction

- Many sub-disciplines: high school, examination, ...
- Can their models be unified?
- Should they be unified?
- Informal discussion, but extensive unpublished work behind it

Is unification possible?

Yes, because all timetabling problems deal with

- Times
- Resources
- Events
- Constraints

Is unification desirable?

For solving probably no, but yes for

- Other software (evaluators, solve platforms)
- Spreading good specification ideas around
- Insight into what timetabling is
- Problems outside or spanning across the usual categories

Existing models

No recent unified models, but plenty of successful sub-discipline models:

- High school timetabling: XHSTT
- Examination timetabling: Toronto
- Nurse rostering: INRC1, INRC2, etc.
- University course timetabling: UniTime/ITC2019
- Sports Scheduling: RobinX

We want something like these, but unified.

Times

Represent the times that events may occur:

- One *time* is a finite set of finite intervals of real time
- Each event can choose from only finitely many specified times
- So time is discretized
- Hierarchy of times (semesters, weeks, days) defined in instance

A generalization of the ITC2019 time model. Issues:

- design elegance
- implementation efficiency

Resources

Represent the participants in events (teachers, rooms, nurses, ...):

- Resource types declared in instance
- Resource attributes declared in instance (room capacity, walking distance etc.)

No issues here.

Events

Represent meetings that resources attend:

- Each event holds a set of times (preassigned or to be assigned)
- Each event also holds a set of *tasks*
- Each task holds a set of resources (preassigned or to be assigned)

No issues here.

Constraints

- Explicitly notated, with hard vs soft, weight, cost function (XHSTT)
- Expression trees for complex cases (e.g. busy weekends, student sectioning)

The main issue: must cover a large range of quirky constraints:

- Walking distance
- Fairness
- ...

while avoiding chaos.

General form of one constraint node

1. *Task selection* selects the set of tasks t such that:
 - t comes from a given set of tasks S
 - t is assigned a resource from a given set of resources R
 - t lies in an event assigned a time from a given set of times T
2. *Determinant function* maps the selected set of tasks to an integer *determinant*; e.g.
 - Number of tasks
 - Total walking distance from each task to the next
 - ...
3. *Cost calculation* maps the determinant to a cost using the usual limits and weights

Reducing verbosity

XML formats are notoriously verbose. This would be a chance to fix that:

- Iterators like ‘for each time t in T , for each resource r in R , ...’
- Smaller improvements like

```
cost="count:2-5 | s20"
```

Conclusion

- Unification is possible
- There are some good reasons to do it
- Is anyone interested?