**MISTA 2013**, Ghent, Belgium (27-30 Aug 2013)

# Initial Results on Fairness in Examination Timetabling

**Ahmad Muklason[1]**
**Dr. Andrew J. Parkes[1], Dr. Barry McCollum[2], Dr. Ender Ozcan[1]**

[1] **Automated Scheduling , Optimisation and Planning (ASAP) Research Group**
**School of Computer Science, The University of Nottingham, Nottingham, NG8 1BB, UK.**
[2] **School of Computer Science, Queen's University, Belfast, BT7 1NN, UK.**

# Outline

- Motivation and Objectives
- Problem formulation
- Data Set
- Method
- New Proposed Approach
- Experimentation
- Results and Discussion
- Future Works

# Motivation and Objectives

- Motivation

  Usual objective function: minimise linier sum of cost.

  Problem: cost associated to entities e.g. students is **unbalanced** -> unfair solution.

- Objectives

  Proposed new approach to produce **fairer solutions**.

- Related works: fairness in computer network, flight landing scheduling, nurse rostering, course timetabling, but no prior work in exams timetabling problems.

# Problem formulation

- Assignment each exam to timeslot such that no two exams with common students assigned to the same timeslot.

(minimise)

$$\frac{\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} C_{ij} W_{|t_j - t_i|}}{S} \tag{2.4}$$

Subject to:

$$\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} C_{ij}\delta(t_i,t_j) = 0, \text{where } \delta(t_i,t_j) = \begin{cases} 1 & \text{if } t_i = t_j \\ 0 & \text{otherwise} \end{cases} \tag{2.5}$$

$$W_{|t_j - t_i|} = \begin{cases} 0, & if\ |t_j - t_i| > 5 \\ 2^{5-|t_j - t_i|}, & otherwise \end{cases}$$

# Problem formulation (cont.)

- Fairness?
  - Max-Min (qualitative measure)
  - Quantitative measure of fairness/inequality in resource allocation:
    - Jain Fairness Index (JFI)
    - Gini Index
- **JFI** is used in the experimentation.

# Data Set

- Carter Dataset

Table 2.4: Carter Dataset

| Problems | Exams | Students | Density | Timeslots |
|---|---|---|---|---|
| CAR91 | 682 | 16925 | 0.13 | 35 |
| CAR92 | 543 | 18419 | 0 .14 | 32 |
| EAR83 | 190 | 1125 | 0.27 | 24 |
| HEC92 | 81 | 2823 | 0.42 | 18 |
| KFU93 | 461 | 5349 | 0.06 | 20 |
| LSE91 | 381 | 2726 | 0.06 | 18 |
| PUR93 | 2419 | 30029 | 0.03 | 42 |
| RYE92 | 486 | 11483 | 0.07 | 23 |
| STA83 | 139 | 611 | 0.14 | 13 |
| TRE92 | 261 | 4360 | 0.18 | 23 |
| UTA92 | 622 | 21266 | 0.13 | 35 |
| UTE92 | 184 | 2749 | 0.08 | 10 |
| YOR83 | 181 | 941 | 0.29 | 21 |

# Method

- Method : Two Phases
  - **Phase 1** : Generating initial feasible solution.
  - **Phase 2** : Improving the quality of initial solution, i.e. minimise soft constraints violation.

# Method (cont.)

- **Phase 1**

  Generating initial feasible solution: Sequential heuristic construction method

1. Generate Graph (**G**) from problem instance.
2. Calculate maximal clique (**MC**) from **G**.
3. Colour each vertex in **MC** with different colour (colour starts from **1**).
4. Order remaining vertices in non-increasing saturation degree.
5. Of remaining vertices, colour each vertex with a feasible colour in the order given by step no. 4 while updating the saturation degree of neighbouring vertices.

# Method (cont.)

- **Phase 2**

  Improving the quality of initial solution, i.e. minimise soft constraints violation: Great Deluge Algorithm

```
1.  set   NIL ← non improvement limit
2.  S   ← initial feasible solution // from previous step
3.  initCost   ← costFunction(S)
4.  BoundaryLevel   ← initCost *0.9
5.  set decayRate  ← BoundaryLevel /10000000
6.  lastMove   ← 0
7.  while ((BoundaryLevel > 0 AND initSolution < BoundaryLevel) OR (lastMove <
    NIL)) do
8.    Apply neighbourhood Heuristic S* on S
9.    Calculate costFunction (S*)
10.    If costF unction(S*) <= costF unction(S) OR (costFunction(S*) <=  BoundaryLevel
    Then
11.   Accept S = S*
12.   BoundaryLevel = B * decayRate
13.   lastMove =0
14.   Else lastMove++
15. end while
```

# New Proposed Approach

- **Prior Approach**

$$LS = \sum_{s=1}^{S} P_s$$

- **Proposed Approach**

$$MS = \sum_{s=1}^{S} f(P_s)$$

Where,

$$P_s = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} P(s,i,j)$$

$$P(s,i,j)) = \begin{cases} W_{|t_j - t_i|} & \text{if student s takes exams i and j} \\ 0 & \text{otherwise} \end{cases}$$

- ✓ **LS** = Linier summation
- ✓ **MS** = Modified summation
- ✓ **Ps** = Total penalty (cost) associated to individual sth student.
- ✓ **S** = number of students.
- ✓ **$W|t_j - t_i|$** = approximation cost.

# New Proposed Approach (cont.)

**Proposed objective functions:**

- **Sum-of–Squares (SoS)**
- **Sum-of-Powers (SoP)**

SoS : $f(P_s) = P_s^2$

SoP(p): $f(P_s) = P_s^p$

# Experimentation

- For each instance in Carter dataset we conduct 20 test runs.
- The fairness of obtained solutions is measured with **Jain Fairness Index (JFI)**.

$$JFI(T) = \frac{\left(\sum_{s=1}^{S} P_s\right)^2}{\left(S * \sum_{s=1}^{S} (P_s)^2\right)} \qquad (2.14)$$

- The trade-off between overall mean penalty and the fairness is evaluated.

# Result

Table 2.5: Experimental result: solutions' fairness produced from LS, SoS, and SoP objective function

| Instance | LS | | SoS | | SoP | | Improv. SoS over LS | | Improv. SoP over LS | |
|---|---|---|---|---|---|---|---|---|---|---|
| | MEAN | JFI | MEAN | JFI | MEAN | JFI | dMean(%) | dJFI (%) | dMean(%) | dJFI (%) |
| CAR91 | 5.46 | 0.33 | 5.56 | 0.35 | 8.33 | 0.39 | 1.83 | 6.06 | 52.56 | 18.18 |
| CAR92 | 4.96 | 0.29 | 4.83 | 0.31 | 6.74 | 0.35 | -2.62 | 6.90 | 35.89 | 20.69 |
| EAR83 | 39.8 | 0.82 | 39.42 | 0.84 | 47.88 | 0.86 | -0.95 | 2.44 | 20.30 | 4.88 |
| HEC92 | 11.24 | 0.49 | 11.34 | 0.52 | 15.32 | 0.58 | 0.89 | 6.12 | 36.30 | 18.37 |
| KFU93 | 16.25 | 0.54 | 16.26 | 0.56 | 20.07 | 0.63 | 0.06 | 3.70 | 23.51 | 16.67 |
| LSE91 | 13.36 | 0.53 | 13.42 | 0.57 | 17.16 | 0.63 | 0.45 | 7.55 | 28.44 | 18.87 |
| PUR93 | 5.84 | 0.35 | 6.06 | 0.38 | 8.62 | 0.44 | 3.77 | 8.57 | 47.60 | 25.71 |
| RYE92 | 9.82 | 0.37 | 9.7 | 0.39 | 16.21 | 0.44 | -1.22 | 5.41 | 65.07 | 18.92 |
| STA83 | 163.08 | 0.91 | 163.12 | 0.91 | 168.01 | 0.93 | 0.02 | 0.00 | 3.02 | 2.20 |
| TRE92 | 8.82 | 0.44 | 8.93 | 0.47 | 11.16 | 0.49 | 1.25 | 6.82 | 26.53 | 11.36 |
| UTA92 | 4.23 | 0.24 | 4.31 | 0.26 | 5.22 | 0.29 | 1.89 | 8.33 | 23.40 | 20.83 |
| UTE92 | 26.54 | 0.79 | 26.51 | 0.8 | 31.59 | 0.81 | -0.11 | 1.27 | 19.03 | 2.53 |
| YOR83 | 39.12 | 0.75 | 38.8 | 0.77 | 48 | 0.77 | -0.82 | 2.67 | 22.70 | 2.67 |

# Discussion & Conclusion

- Both SoS and SoP objective function can produce fairer solutions for all problem instances.
- SoP can produce greater amount of fairer solution than SoS.
- SoS compensates smaller increase in average penalty than SoP, even can produce smaller average for **CAR92, EAR83, RYE92, UTE92**, and **YOR83.**

# Future Works

- Taking into consideration the quality and fairness of solution from the perspective of other stakeholders such as invigilators, lecturers, and estates, etc.
- Propose more realistic formulations.

- Thanks you ☺
- Q&A

# Key References


- McCollum, B. (2007). A perspective on bridging the gap between theory and practice in university timetabling. Paper presented at the Proceedings of the 6th international conference on Practice and theory of automated timetabling VI, Brno, Czech Republic.