Hyper-Heuristics for Examination Timetabling with Fairness

A. Muklason[1], A.J. Parkes[1], B. McCollum[2], E. Özcan[1]

[1]University of Nottingham, [2]Queen's University of Belfast

{*abm, ajp, exo*}*@cs.nott.ac.uk, b.mccollum@qub.ac.uk*

August 22, 2014

Overview

Introduction

- Well-known as challenging real world optimisation problems (NP-Hard).
- Could be considered as an assignment problems of exams into limited number resources (i.e. time-slots and rooms) subject to some restricted constrains.
- **Hard constraints**, e.g. Clash-free timetable (no student has to sit more than one exam in the same time). Determine the *feasibility* of solution (i.e. timetable).
- **Soft constraints**, e.g. spread out the timetable as evenly as possible. Determine the *quality* of solution.
- State-of-the-art approaches to solving examination timetabling problems can be found in [Qu et. al., 2009].

Examination Timetabling Problem Dataset

Two most-researched public examinations timetabling problem domains:

Toronto Dataset [Carter et. al., 1996]

Consist of 13 datasets. It is referred as un-capacitated examination timetabling problems i.e. Room capacity is not taken into consideration. Proximity cost penalty was introduced.

International Timetabling Competition 2007 (ITC 2007) Dataset [McCollum et. al., 2012]

It is referred as capacitated examination timetabling problems. More complex hard and soft constraints reflecting the real examination timetabling problems.

Toronto Benchmark Dataset

Problem	Time-slots	Exams	Students	C.Density
car92	32	543	18419	0.14
car91	35	682	16925	0.13
ears83l	24	190	1125	0.27
hec92l	18	81	2823	0.42
kfu93	20	461	5349	0.06
lse91	18	381	2726	0.06
pur93l	42	2419	30032	0.03
rye92	23	486	11483	0.08
sta83l	13	139	611	0.14
tre92	23	261	4360	0.18
uta92l	35	622	21266	0.13
ute92	10	184	2750	0.08
yor83l	21	181	941	0.29

Table : Toronto Benchmark Dataset

ITC 2007 Benchmark Dataset

Problem	Time-slots	Exams	Students	Rooms	C.Density
Exam1	54	607	7891	7	5.05
Exam2	40	870	12743	49	1.17
Exam3	36	934	16439	48	2.62
Exam4	21	273	5045	1	15.0
Exam5	42	1018	9253	3	0.87
Exam6	16	242	7909	8	6.16
Exam7	80	1096	14676	15	1.93
Exam8	80	598	7718	8	4.55

Table : ITC 2007 Benchmark Dataset

Toronto Exams Problem Formulation

Objective Function:Minimise

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

Where $P_{|t_j-t_i|} = 2^{5-|t_j-t_i|}$ and $\mid t_j - t_i \mid \in \{1,2,3,4,5\}$

Subject to:

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

N : the number of examinations.
C_{ij}: the number of students sit for both examination i and j.
t_i/t_i :assigned time-slot for examination i/j.
S:the total number of students.

ITC 2007 Exams Problem Formulation

Hard Constraints:

- No student sits more than one examination at the same time;
- The capacity of individual rooms is not exceeded at any time throughout the examination session;
- Period Lengths are not violated; Satisfaction of period related hard constraints e.g. ExamA After ExamB;
- Satisfaction of room related hard constraints e.g. ExamA must use Room 101.

ITC 2007 Exams Problem Formulation

Objective Function: Minimise

$$\sum_{s \in S}(w^{2R} C_s^{2R} + w^{2D} C_s^{2D} + w^{PS} C_s^{PS}) + w^{NMD} C^{NMD} + w^{FL} C^{FL} + C^P + C^R$$

Where,
C_s^{2R} = two exams in a row penalty for student s.
C_s^{2D} = two exams in a day penalty for student s.
C_s^{PS} = period spread penalty for student s.
C^{NMD} = No mixed duration penalty.
C^{FL} = Front-Load penalty.
C^P = soft period penalty.
C^R = soft room penalty.

Fairness In Examination Timetabling Problems

Our Pilot survey on examination timetable preference of university students reveals two important feedback that lack of attention in the state-of-the-art research in Exam timetabling:

- Fairness does matter, especially among students within the same course.
- Exam hardness, hard exams require more time for preparation.

This feedback suggest new model and consequently need new algorithm. In this work we focus on tackling fairness issue.

Fairness In Examination Timetabling Problems

- Important, since exams contribute significantly to students academic achievement.
- No previous works investigated fairness in exams timetabling problems

Research Objective

- How to make balance between quality of the overall timetable and fairness to individual students?

Fairness Formulation

To enforce fairness, we modify the objective function formulation. Sum of Power (SOP) is used instead of standard linear summation. Let s is number of students, x is any positive integer greater than 1, and $p_1, p_2, ..., p_s$ be the total penalty received by students respectively. This objective function will make $F(3, 4, 3)$, which is fairer, preferable than $F(1, 6, 3)$ although both linear summation are the same.

$$F(p_1, p_2, ..., p_s) = (p_s)^x \tag{1}$$

Fairness Formulation

To measure fairness, Jain Fairness Index is used.

Jain Fairnes Index (JFI):

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

S : the total number of students
P_s: total penalty suffered student s
JFI value is bounded between 0 and 1, in which 0 is completely unfair while 1 is completely fair.

Hyper-heuristics

Hyper-heuristics, a well-known effective approach for solving many combinatorial optimisation problems, are implemented within the HyFlex framework.
After an initial feasible solution is constructed, an extended great deluge algorithm is applied as a strategy to select predefined low-level heuristics in order to minimise an objective function that includes the standard soft constraints, but also includes fairness measured by Jain Fairness Index.

Initial Feasible Solution Construction

- An initial feasible solution is constructed by using graph colouring based heuristic ordering.
- Basic Principle: The most 'difficult' (in term of the availability of feasible time-slots and room) exam is assigned to time-slot and room first.

Heuristic Ordering

Heuristic Ordering used:

Heuristic	Explanation
Saturation Degree (SD)	Increasing order based on remaining feasible time-slot
Largest Weighted Degree (LWD)	Decreasing order based on the number of exams in conflict weighted with its number of the enrolled students
Largest Enrolment (LE)	Decreasing order based on the number of the enrolled students
Largest Colour Degree (LCD)	Decreasing order based on the number of already assigned to time-slot exams in conflict

Table : Heuristic Ordering List

Initial Feasible Construction Algorithm

```
1: G = graph generated from problem instance
2: k = 0
3: T = number of time-slots
4: repeat
5:     M = maximal Clique of graph G
6:     t = 1
7:     for all exam e in M do
8:         Assign exam e to time-slot t
9:         t = t + 1
10:    end for
11:    k = size of M
12:    U = ordered list of unassigned exam based on saturation degree
13:    while U is not empty do
14:        e= the first exam in U
15:        for i = 1; ; i = i + 1 do
16:            if i is feasible then
17:                assign exam e to time-slot i
18:                if i > k then
19:                    k = i
20:                end if
21:                break
22:            end if
23:        end for
24:        remove exam e from U
25:        update saturation degree of exams adjacent with e
26:    end while
27: until k ≤ T
```

Algorithm 1: Pseudo-code for generating initial feasible of Toronto Instances

Initial Feasible Solution Construction

Initial Feasible Solution Construction for ITC 2007 Approach:

- Exams are ordered based on different hybridisation of some heuristic ordering .
- Each unscheduled exam is given a difficulty index, in which it is measured based on its difficulty based on above different hybridisation of some heuristic ordering.

Heuristic Ordering

Heuristic Ordering List of Exams:

Heuristic Ordering List	Explanation
L1	LWD+LE : exams are ordered based on largest weighted degree first, a tie is broken by largest enrolment first ordering
L2	SD+LCD : same as above, but ordered based on saturation degree and largest colour degree
L3	LCD+SD : same as above, but ordered based on largest colour degree and saturation degree
L4	LE+LWD : same as above, but ordered based on largest enrolment and largest weighted degree

Table : Ordering List of Exams

Difficulty Index

Given four ordering list : $L1, L2, L3, L4$. The difficulty index (DI)of each exam i i.e. e_i is calculated as bellow:

Difficulty Index:

$$DI(e_1) = \sum_{l=1}^{4} I_{li}$$

I_{li} : the order (position) of exam e_i in ordering list i i.e. L_i

Initial Feasible Construction Algorithm

1: X = list of exams requiring exclusive room
2: **for all** exam e in X **do**
3: assign exam e to randomly chosen feasible time-slot t and feasible room r
4: **end for**
5: C = list of exams with coincidental exams
6: **for all** exam e in X **do**
7: assign exam e to randomly chosen feasible time-slot t and feasible room r.
8: **end for**
9: $L1$ = list of exams with LWD+LE ordering
10: $L2$ = list of exams with SD+LCD ordering
11: $L3$ = list of exams with LCD+SD ordering
12: $L4$ = list of exams with LE+LWD ordering
13: **while** $L1$ is not empty **do**
14: calculate the difficulty index of the first exam in each list
15: e = exam with the minimum difficulty index, choose randomly if there is more than one exams with the same difficult index
16: assign exam e to randomly chosen feasible time-slot t and feasible room r.
17: remove exam e from $L1, L2, L3, L4$
18: update saturation degree and colour degree of exams in conflict with exam e
19: **end while**

Algorithm 2: Pseudo-code for generating initial feasible of ITC 2007 Instances

Improvement Stage

- Hyper-heuristic approach with extended great deluge algorithm.
- Work upon low-level heuristics space rather than solution space.

Low-Level Heuristics

Low-Level Heuristic for Toronto Instance

Heuristic Ordering List	Explanation
Random Move	choose an exam randomly, and assign to random new time-slot
Swap Two	choose two exams randomly, and swap the time-slot
Kempe Chain	choose two time-slots randomly. Generate one pair Kempe chain of those two time-slots. Swap the time-slot of exams in the Kempe chain

Table : Low-Level Heuristic for Toronto Instance

Low-Level Heuristics

Low-Level Heuristic for ITC 2007 Instance

Heuristic Ordering List	Explanation
Swap Exam	two exams are chosen randomly, then swap their time-slot and room
Change Period	choose an exam randomly, assign to randomly chosen new time-slot
Change Room	choose an exam randomly, assign to randomly chosen new room
Swap Room	two exams are chosen randomly, then swap their room
Change Period and Room	choose an exam randomly, assign to randomly chosen new time-slot and room
Swap Period	two exams are chosen randomly, then swap their time-slot

Table : Low-Level Heuristic for ITC 2007 Instance

Extended Great Deluge Hyper-Heuristic

1: Input: $NLLH$ =number of low level heuristics, TL=time limit
2: construct initial solution S
3: calculate cost function of initial solution $f(S)$
4: set initial boundary level $B = f(S)$
5: set initial decay rate $dRate$
6: **while** not exceed TL **do**
7: choose a low level heuristic, llh randomly from set of low level heuristics
8: apply llh on S to generate new solution S^*
9: calculate cost function of new solution $f(S^*)$
10: **if** $f(S^*) \leq f(S)$ OR $f(S^*) \leq B$ **then**
11: accept $S = S^*$
12: **end if**
13: **if** No improvement within given time WT **then**
14: reset $B = f(S)$
15: reset decay rate $dRate$ = secondary cooling parameter
16: **end if**
17: **end while**

Algorithm 3: Pseudo-code of Extended Great Deluge Hyper-Heuristic

Experimental Results

Problem	LS		SoS		SoP	
	Mean	JFI	Mean	JFI	Mean	JFI
CAR91	5.46	0.33	5.56	0.35	8.33	0.39
CAR92	4.96	0.29	4.83	0.31	6.74	0.35
EAR83	39.8	0.82	39.42	0.84	47.88	0.86
HEC92	11.24	0.49	11.34	0.52	15.32	0.58
KFU93	16.25	0.54	16.26	0.56	20.07	0.63
LSE91	13.36	0.53	13.42	0.57	17.16	0.63
PUR93	5.84	0.35	6.06	0.38	8.62	0.44
RYE92	9.82	0.37	9.7	0.39	16.21	0.44
STA83	163.08	0.91	163.12	0.91	168.01	0.93
TRE92	8.82	0.44	8.93	0.47	11.16	0.49
UTA92	4.23	0.24	4.31	0.26	5.22	0.29
UTE92	26.54	0.79	26.51	0.8	31.59	0.81
YOR83	39.12	0.75	38.8	0.77	48	0.77

Table : Experimental Result on Toronto Benchmark Dataset

Experimental Results

Problem	LS		SoS		SoP	
	Mean	JFI	Mean	JFI	Mean	JFI
Exam1	29257	0.56	31328	0.59	NA	NA
Exam2	37208	0.20	NA	NA	NA	NA
Exam3	122098	0.37	NA	NA	NA	NA
Exam4	50420	NA	NA	NA	NA	NA
Exam5	41064	0.62	NA	NA	NA	NA
Exam6	150628	0.44	NA	NA	NA	NA
Exam7	63672	NA	NA	NA	NA	NA
Exam8	124902	NA	NA	NA	NA	NA

Table : Experimental Result on ITC 2007 Benchmark Dataset

Conclusion

- The proposed constructive heuristic algorithm able to generate feasible initial solutions for all problem instances in Toronto and ITC 2007 Problem domain.
- SoP objective function can produce fairer solutions for all problem instances.
- There is trade-off between efficiency (core objective function) and fairness.

Future Works

New model formulation for exam timetabling problems that cope with fairness and exam hardness. Tackling the trade-off between efficiency core objective function and fairness in multi-objective optimisation fashion. Low-level heuristics as well as higher strategy to choose the low-level heuristics within Hyper-heuristics will be developed to cope with the new model.

References

Qu, R., Burke, E. K., Mccollum, B., Merlot, L. T., and Lee, S. Y. (2009)

A survey of search methodologies and automated system development for examination timetabling

*Journal of Scheduling*12(1),55-89.

Carter, M. W., Laporte, G., & Lee, S. Y. (1996)

Examination timetabling: Algorithmic strategies and applications

*Journal of the Operational Research Society*47(3), 373-383.

McCollum, B., McMullan, P., Parkes, A. J., Burke, E. K., & Qu, R. (2012).

A new model for automated examination timetabling

*Ann Oper Res*194(1), 291-315.

The End

Thank You
Questions?