International Conference on Electrical Engineering and Computer Science 2018
(ICECOS 2018)
02-04 October 2018, Bangka Island, Province of Bangka-Belitung, Indonesia

Automated Examination Timetabling Optimization Using Greedy-Late Acceptance Hyper-heuristic Algorithm

Ahmad Muklason
Data Engineering and Business Intelligence Lab.
Department of Information Systems
Faculty of Information and Communication Technology
Institut Teknologi Sepuluh Nopember (ITS)
Surabaya

Outline

- Introduction
- Problem Formulation
- Method
- Dataset
- Experimental Results
- Conclusion

Introduction

- Administrator so stressful preparing the exam timetable.
- Some students complaining got two, even three exams in row.
- Theoretically, exam timetabling problem is NP-hard problem.
- Contributions: new public dataset, new algorithmic approach: greedy + late acceptance hyper-heuristics

Problem Formulation

Decision Variable:

$$X_{it} \begin{cases} 1, \text{ if } i^{th} \text{ course is scheduled at } t^{th} \text{ timeslot} \\ 0, \text{ otherwise} \end{cases}$$

Hard Constraints:

$$\sum_{i=1}^{n} \sum_{t=1}^{k} x_{it} = 1$$

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} c_{ij} v_{ij} = 0, \; v_{ij} \begin{cases} 1, \text{ if } t_i = t_j \\ 0, \text{ otherwise} \end{cases}$$

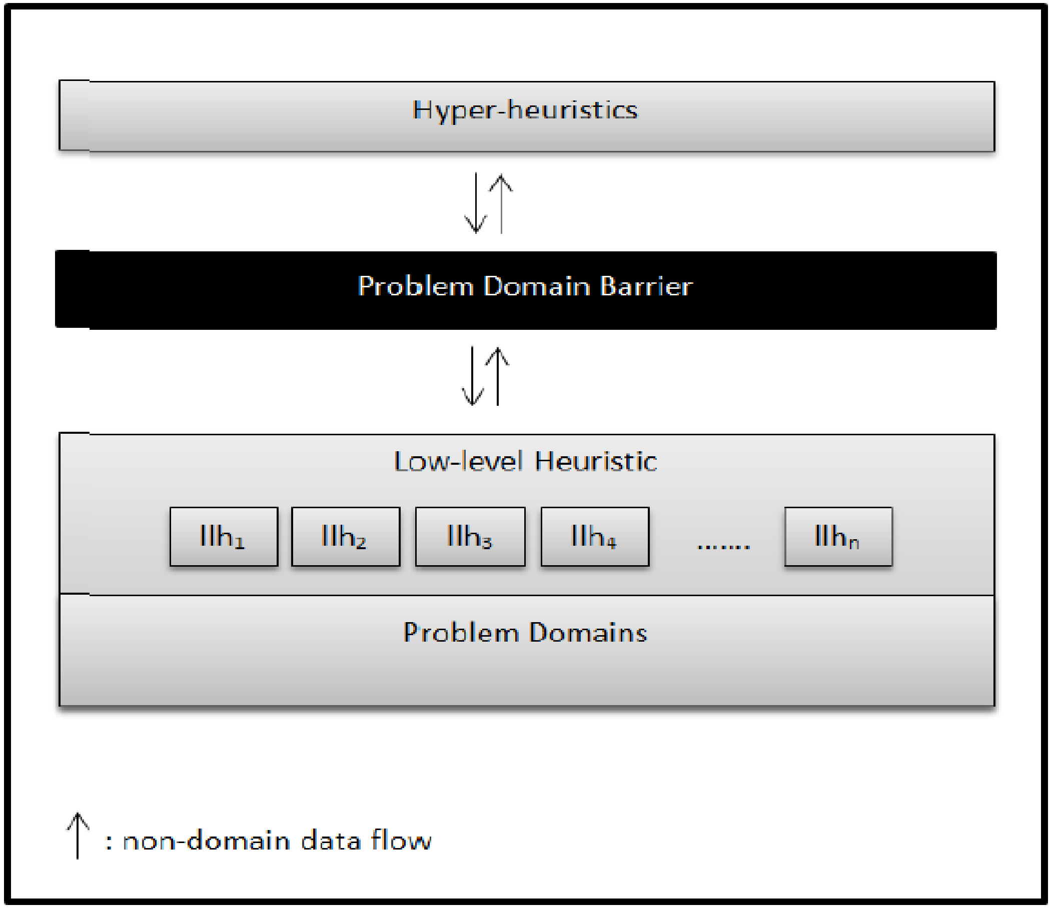
$$\sum_{i=1}^{n} s_i x_{it} \leq \mathbf{R}$$

Objective Function:

$$P = \frac{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} c_{ij} w_{|t_j - t_i|}}{M}, \; w_{|t_j - t_i|} \begin{cases} 2^{5-|t_j - t_i|}, \text{ if } 1 \leq |t_j - t_i| \leq 5 \\ 0, \text{ if } |t_j - t_i| > 5 \end{cases}$$

Method

New algorithmic approach, i.e. **hyper-heuristics**



Method (cont.)

New algorithmic approach, i.e. **hyper-heuristics**

- **Stage 1: Generate initial feasible solution**

  Greedy Algorithm: The most conflicting exam is scheduled first.

- **Stage 2: Improve the initial solution**

  Hyper-heuristics:

  1) **Low-level heuristic selection strategy: Simple Random {move, swap}**
  2) **Move acceptance strategy: Late acceptance**

Method (cont.)

New algorithmic approach, i.e. **hyper-heuristics**

Move acceptance strategy: Late acceptance

Supposed, the algorithm is run within i^{th} iteration.

The new solution from i^{th} iteration is compared with the solution recorded from $(q\%i)^{th}$ iteration instead of with the best solution found so far (as in hill climbing).

q is the length of list (**queue**) recording solutions from the last q iterations.

0	1	2	...	...	...	...	...	...	q

Dataset

	Dataset #1	Dataset #2
# students	567	519
# exams	32	32
# timeslots	15	15
Max Room capacity	160	160

The data is real-world examination timetabling problem from Dept. of Information Systems, ITS.

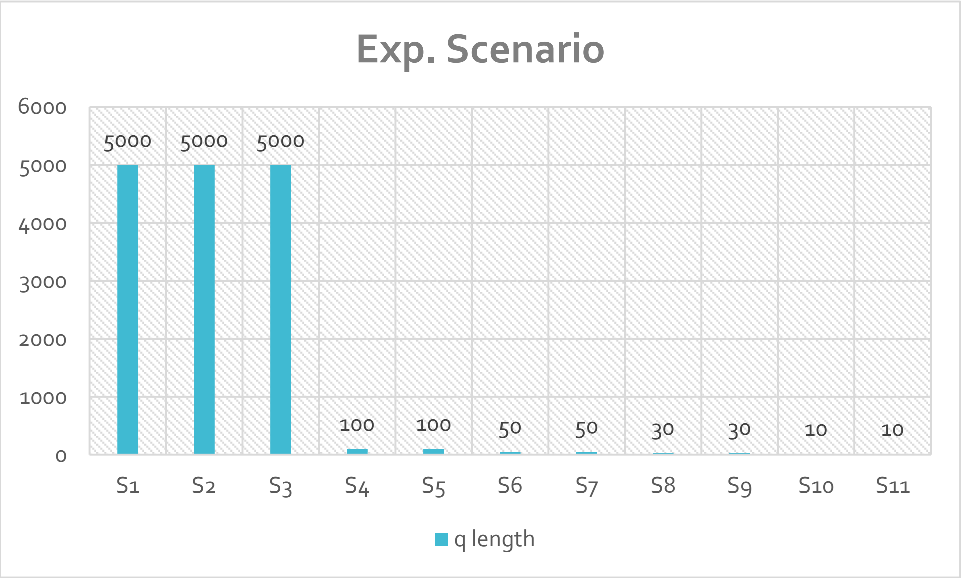
Experimental Results

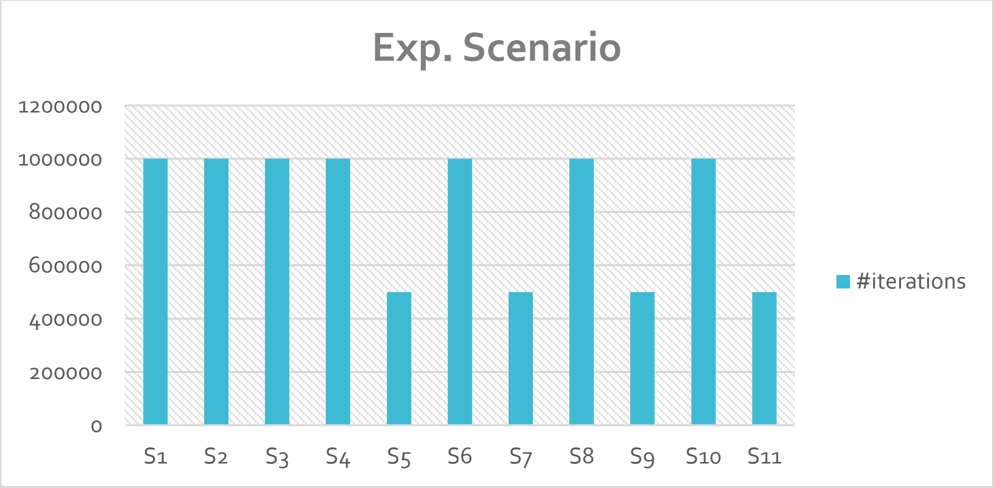
- Experimental Scenarios (LLH):

Scenario	Low-level Heuristics
s1	Simple Random (Move and Swap)
s2	Move
s3	Swap
s4	Simple Random (Move and Swap)
s5	Simple Random (Move and Swap)
s6	Simple Random (Move and Swap)
s7	Simple Random (Move and Swap)
s8	Simple Random (Move and Swap)
s9	Simple Random (Move and Swap)
s10	Simple Random (Move and Swap)
s11	Simple Random (Move and Swap)

Experimental Results

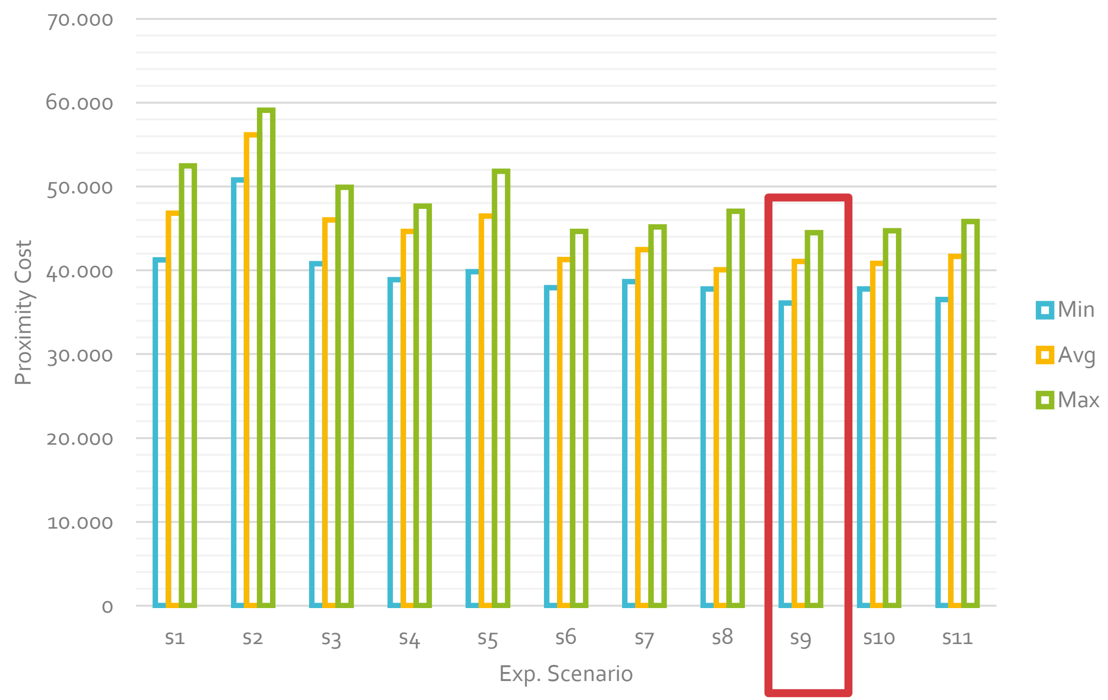
- Experimental Scenarios (q length and # iterations):





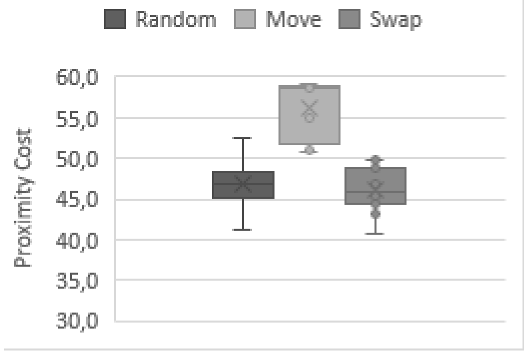
Experimental Results (cont.)

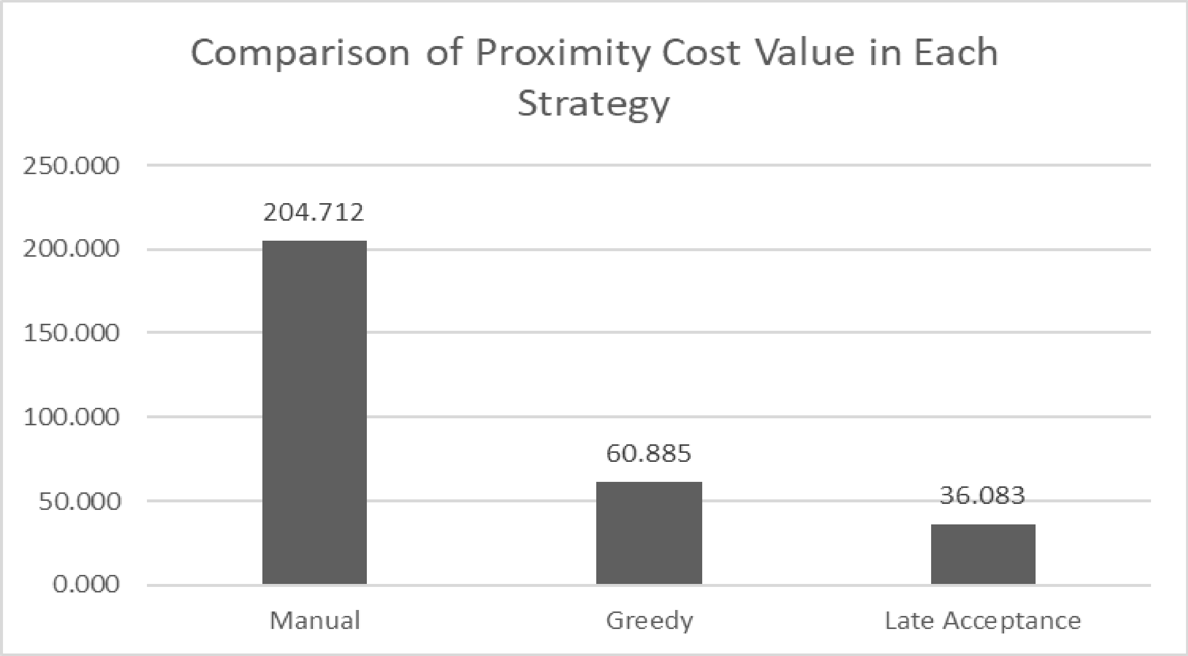
- Summary from 11 runs



Experimental Results (cont.)

- Exp. Results (Best Results)





Conclusion

- The proposed algorithm: Greedy-Late Acceptance Hyper-heuristic Algorithm could solve the problem effectively.
- Compared to the timetable generated manually:
  - Much more efficient: **5 mins** compared to (normally) **two weeks**
  - Much better quality (in term of proximity cost that should minimised): **36** compared to **205**

Conclusion (cont.)

- Future works recommendations:
  - Comparing with other **sophisticated algorithms**.
  - Should be tested over **bigger and varied data sets**.
  - Should be tested over different problem domains/ problem instances.
  - Should include **invigilator scheduling**.

Thank you ☺