Hybridising Heuristics within a Graph based Hyper-Heuristic Framework

Dr Rong Qu, Associate Professor
ASAP Group, The University of Nottingham
rong.qu@nottingham.ac.uk
http://www.cs.nott.ac.uk/~pszrq

NATCOR – Heuristics and Approximate Algorithms
Nottingham, April, 2016
Assigning a set of exams into limited timeslots satisfying
- **Hard constraints**: cannot be violated
- **Soft constraints**: desired
- **Quality of solutions**: objective function

- **Events**
- **Timeslots**
- **Rooms**
- **Etc.**
Educational Timetabling

- Important activities in all universities
  - **Hard constraints**: No events for students at the same time
  - **Soft constraints**: Spread students’ events

- State-of-the-art: different “tailor-made”, “fine-tuned” techniques
  - Graph heuristics, constraint based techniques
  - Meta-heuristics, multi-criteria
  - Recent developments:
    - **hybrid** techniques, hyper-heuristics, VNS, ILS, GRASP, adaptive techniques, etc.

---


---

NATCOR – Heuristics and Approximate Algorithms, Nottingham, April 2016
Educational Timetabling

- Carter, Laporte & Lee (1996): exam timetabling instances
  - Hard constraint: conflicts between exams
  - Objective function: min time slots (graph colouring)
  - Soft constraints: spread out exams over time slots
  - Objective function: \( C(t) = \left( \sum_{s=0}^{4} w_s N_s \right) / S \)

- Meta-heuristic Network (2000): course timetabling instances
  - Hard constraints: exams conflicts, room features
  - Soft constraints: minimise only one class a day, class in the last slot of a day, more than two classes in a row
  - Objective function: min sum of the costs for soft constraints
  - New benchmark:
A Graph Based Hyper-heuristic

- Hyper-heuristics: **Heuristics** that choose **heuristics**
  - **High level**: Meta-heuristics, Choice function, CBR, etc.
  - **Low level**: moving strategies, constructive heuristics, etc.

- **Aim of hyper-heuristic**
  - Explore general techniques for wider problems
  - High level search doesn’t look into domain knowledge

- **Applications**
  - bin packing, educational timetabling, personal scheduling, etc.

A Graph Based Hyper-heuristic

- **High level search**: Any meta-heuristics
  - Search for lists of **low level heuristics** to construct solutions
  - **Low level heuristics**: order events by how *difficult* to schedule them
    - Saturation Degree: least available slots
    - Colour Degree: most conflicted with those scheduled
    - Largest Degree: most conflicted with the others
    - Largest Weighted Degree: LD + students
    - Largest Enrolment: students enrolled
    - Random Ordering: brings randomness
    - Bin packing: best fit, first fit
A Graph Based Hyper-heuristic

exams

| e1 | e2 | e3 | e4 | e5 | e6 | e7 | e8 | e9 | e10 | e11 | e12 | ... |

Heuristic list

SD  SD  LD  CD  LE  SD  SD  LW  SD  LD  CD  RO  ...

order of exams

| e1 | e9 | e3 | e26 | e25 | e6 | e17 | e28 | e19 | e10 | e31 | e12 | ... |

slots

| e1 | e9 | e3 | e26 | e25 |   |   |   |   |   |   |   |   |
A Graph Based Hyper-heuristic

exams

Heuristic list

order of exams

slots

NATCOR – Heuristics and Approximate Algorithms, Nottingham, April 2016
A Graph Based Hyper-heuristic

exams

| e2 | e4 | e5 | e7 | e8 | e11 | e12 | ... |

Heuristic list

| SD | SD | LD | CD | LE | SD | SD | LW | SD | LD | CD | RO | ... |

order of exams

| e5 | e32 | e19 | e22 | e13 | e31 | e12 | e7 | e2 | e15 | e27 | e12 | ... |

slots

| e1 | e9 | e3 | e6 | e19 | e26 | e25 | e28 | e17 | e10 | e5 | e13 | e32 | e19 | e13 |
A Graph Based Hyper-heuristic

- Graph based Hyper-heuristics (GHH) Framework
  - **Search space**: permutations of graph heuristics, rather than actual solutions
  - **Moving operator**: randomly change two heuristics in the heuristic list within a local search
  - **Objective function**: maps heuristic lists to penalty of timetables constructed

- Further investigations
  - **Role** of different high / low level heuristics (ILS, TS, SDM, VNS)
  - Characteristics of **heuristic** search space
  - Search in **two** search spaces
Which High Level Heuristics?

- High level search methods
  - Iterated Local Search
  - Tabu Search
  - Steepest Descent
  - Variable Neighbourhood Search

- Objective function
  - heuristic lists → penalties (costs of timetables constructed)
  - “Walks” are allowed. Why?
## Which High Level Heuristics?

<table>
<thead>
<tr>
<th></th>
<th>car91</th>
<th>car92</th>
<th>ear83</th>
<th>hec92</th>
<th>kfu93</th>
<th>ise91</th>
<th>sta83</th>
<th>tre92</th>
<th>ute92</th>
<th>uta93</th>
<th>yor83</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SDM</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>best</td>
<td>5.44</td>
<td>4.87</td>
<td>35.54</td>
<td>12.59</td>
<td>15.25</td>
<td>13.01</td>
<td>160.3</td>
<td>9.01</td>
<td>31.77</td>
<td>3.61</td>
<td>42.77</td>
</tr>
<tr>
<td>avg</td>
<td>6.18</td>
<td>5.3</td>
<td>36.8</td>
<td>12.74</td>
<td>15.63</td>
<td>13.51</td>
<td>163.7</td>
<td>9.37</td>
<td>32.6</td>
<td>4.5</td>
<td>43.6</td>
</tr>
<tr>
<td>time</td>
<td>15367</td>
<td>8001</td>
<td>584</td>
<td>22</td>
<td>2502</td>
<td>1722</td>
<td>69</td>
<td>1597</td>
<td>87</td>
<td>8018</td>
<td>426</td>
</tr>
<tr>
<td><strong>ILS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>best</td>
<td>5.3</td>
<td>4.77</td>
<td>38.39</td>
<td>12.72</td>
<td>15.09</td>
<td>12.72</td>
<td>159.2</td>
<td>8.74</td>
<td>30.32</td>
<td>3.32</td>
<td>40.24</td>
</tr>
<tr>
<td>avg</td>
<td>6.01</td>
<td>5.18</td>
<td>39.58</td>
<td>13.01</td>
<td>15.35</td>
<td>13.1</td>
<td>161.6</td>
<td>8.92</td>
<td>31.3</td>
<td>4.01</td>
<td>43.15</td>
</tr>
<tr>
<td>time</td>
<td>17334</td>
<td>8200</td>
<td>617</td>
<td>31</td>
<td>2629</td>
<td>1832</td>
<td>73</td>
<td>1638</td>
<td>100</td>
<td>10464</td>
<td>527</td>
</tr>
<tr>
<td><strong>TS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>best</td>
<td>5.43</td>
<td>4.94</td>
<td>38.19</td>
<td>12.36</td>
<td>15.97</td>
<td>13.25</td>
<td>165.7</td>
<td>8.87</td>
<td>32.12</td>
<td>3.52</td>
<td>41.3</td>
</tr>
<tr>
<td>avg</td>
<td>6.3</td>
<td>5.34</td>
<td>45.56</td>
<td>14.6</td>
<td>19.55</td>
<td>14.29</td>
<td>169.1</td>
<td>9.67</td>
<td>37.02</td>
<td>4.38</td>
<td>47.97</td>
</tr>
<tr>
<td>time</td>
<td>20393</td>
<td>9111</td>
<td>649</td>
<td>32</td>
<td>2768</td>
<td>1970</td>
<td>80</td>
<td>1800</td>
<td>100</td>
<td>10464</td>
<td>527</td>
</tr>
<tr>
<td><strong>VNS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>best</td>
<td>5.4</td>
<td>4.7</td>
<td>37.29</td>
<td>12.23</td>
<td>15.1</td>
<td>12.71</td>
<td>159.3</td>
<td>8.67</td>
<td>30.23</td>
<td>3.56</td>
<td>43</td>
</tr>
<tr>
<td>avg</td>
<td>6.1</td>
<td>5.1</td>
<td>38.63</td>
<td>12.72</td>
<td>15.24</td>
<td>13.06</td>
<td>163.3</td>
<td>8.88</td>
<td>31.7</td>
<td>4.05</td>
<td>43.93</td>
</tr>
<tr>
<td>time</td>
<td>16321</td>
<td>8107</td>
<td>672</td>
<td>42</td>
<td>2531</td>
<td>1653</td>
<td>47</td>
<td>1721</td>
<td>677</td>
<td>9210</td>
<td>501</td>
</tr>
</tbody>
</table>

NATCOR – Heuristics and Approximate Algorithms, Nottingham, April 2016
Which High Level Heuristics?

<table>
<thead>
<tr>
<th>Heuristics</th>
<th>s1</th>
<th>s2</th>
<th>s3</th>
<th>s4</th>
<th>s5</th>
<th>m1</th>
<th>m2</th>
<th>m3</th>
<th>m4</th>
<th>m5</th>
<th>large</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDM best</td>
<td>7</td>
<td>8</td>
<td>3</td>
<td>6</td>
<td>10</td>
<td>368</td>
<td>100%</td>
<td>367</td>
<td>356</td>
<td>195</td>
<td>100%</td>
</tr>
<tr>
<td>SDM avg</td>
<td>10.8</td>
<td>15.6</td>
<td>5</td>
<td>11.8</td>
<td>12.2</td>
<td>382.5</td>
<td>100%</td>
<td>383</td>
<td>374.5</td>
<td>194.5</td>
<td>100%</td>
</tr>
<tr>
<td>SDM time</td>
<td>15</td>
<td>38</td>
<td>10</td>
<td>8</td>
<td>30</td>
<td>3823</td>
<td>3672</td>
<td>3752</td>
<td>3637</td>
<td>1989</td>
<td>4013</td>
</tr>
<tr>
<td>ILS best</td>
<td>6</td>
<td>9</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>373</td>
<td>461</td>
<td>375</td>
<td>374</td>
<td>172</td>
<td>1132</td>
</tr>
<tr>
<td>ILS avg</td>
<td>8.8</td>
<td>13.2</td>
<td>5.4</td>
<td>7.6</td>
<td>12</td>
<td>375</td>
<td>480.5</td>
<td>377.5</td>
<td>380.5</td>
<td>179.7</td>
<td>1144</td>
</tr>
<tr>
<td>ILS time</td>
<td>32</td>
<td>47</td>
<td>15</td>
<td>11</td>
<td>23</td>
<td>3656</td>
<td>3018</td>
<td>3382</td>
<td>3451</td>
<td>1822</td>
<td>3811</td>
</tr>
<tr>
<td>TS best</td>
<td>11</td>
<td>11</td>
<td>5</td>
<td>11</td>
<td>16</td>
<td>496</td>
<td>533</td>
<td>460</td>
<td>529</td>
<td>214</td>
<td>1164</td>
</tr>
<tr>
<td>TS avg</td>
<td>12.2</td>
<td>16.4</td>
<td>9.2</td>
<td>12.2</td>
<td>18.2</td>
<td>511.5</td>
<td>533</td>
<td>80%</td>
<td>468</td>
<td>539</td>
<td>236</td>
</tr>
<tr>
<td>TS time</td>
<td>12</td>
<td>18</td>
<td>9</td>
<td>7</td>
<td>19</td>
<td>3326</td>
<td>2996</td>
<td>3160</td>
<td>3280</td>
<td>1650</td>
<td>3564</td>
</tr>
<tr>
<td>VNS best</td>
<td>7</td>
<td>12</td>
<td>4</td>
<td>6</td>
<td>6</td>
<td>346</td>
<td>433</td>
<td>359</td>
<td>370</td>
<td>156</td>
<td>1148</td>
</tr>
<tr>
<td>VNS avg</td>
<td>10</td>
<td>14.8</td>
<td>5.2</td>
<td>8</td>
<td>10.6</td>
<td>365</td>
<td>443</td>
<td>369.5</td>
<td>377.5</td>
<td>165.5</td>
<td>1148</td>
</tr>
<tr>
<td>VNS time</td>
<td>32</td>
<td>45</td>
<td>16</td>
<td>10</td>
<td>30</td>
<td>3920</td>
<td>3723</td>
<td>3856</td>
<td>3667</td>
<td>2013</td>
<td>4079</td>
</tr>
</tbody>
</table>

- **Similar** performance within GHH framework (same total no. of evaluations, same initials, etc.), ILS and VNS are slightly better
- Results are comparable to state-of-the-art approaches on both course and exam benchmark problems
Within the GHH framework

- Different subsets of graph heuristics (SD+LD, SD+LWD, SD+LE, SD+LWD+CD, etc.)
  
- With a limited computational time: SD + LWD performed the best

- With more graph heuristics: Longer time given, the better the results
- \( h' \) (\( l \): length of the sequence, \( h \): number of graph heuristics)
- Larger search space, more solutions sampled

- Random ordering also contributes

Which Low Level Heuristics?
Two Search Spaces

- Search space of high level heuristics: permutations of low level heuristics
- Solution space of problem: actual solutions
- Are all the solutions in solution space reachable?
  - GHH: search is upon heuristics, not solutions

Two Search Spaces

<table>
<thead>
<tr>
<th>Representation</th>
<th>Heuristic space</th>
<th>Solution space</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size (Upper Bound)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Neighborhood Operator</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Objective Function</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Representations of graph heuristics**

- **Size (Upper Bound)**: length of the sequence, number of graph colouring heuristics
- **t**: number of timeslots, **e**: number of events

**Neighbor- hood Operator**

- Randomly change two heuristics
- Move events in one timeslot to other timeslots

**Objective Function**

- Penalty of timetables constructed by heuristic sequence
- Penalty of timetables, or difference of costs caused by moving events in the timetable

NATCOR – Heuristics and Approximate Algorithms, Nottingham, April 2016
Search in Two Spaces

- With one move
  - Local search approaches
  - Graph based hyper-heuristics

One bit different
One part different (from different heuristic lists)
Search in Two Spaces

- Local search based algorithms
  - Move within limited search areas
  - Easily stuck to local optima: different mechanisms developed
  - Search attracted within limited parts of search space

- GHH
  - Change the way of building the solutions at a high level
  - Search space of heuristics -> solutions far from each other in the solution space
  - Key feature: coverage of the solution space
  - GHH vs. VNS?
Search in Two Spaces

- **Hybridisation** with greedy local search
  - Coverage of solution space: Results greatly improved!
  - **Diversification** by GHH in the heuristic space, vs.
  - **Intensification** by local search in the solution space
  - Hybrid GHH vs. Memetic Algorithms

NATCOR – Heuristics and Approximate Algorithms, Nottingham, April 2016
Heuristic hybridisations in GHH

- Knowledge: best solutions → good heuristic hybridisation

- I - Random GHH (SD+LWD, SD+LE, SD+LD)
  - A large collection of different heuristic sequences

- II - Analyse the best 5% heuristic sequences
  - Rates of hybridisation at different parts of heuristic sequences
  - Patterns of hybridizations in the best sequences

---

Extension I: Adaptive GHH

- Heuristic hybridisations in GHH
  - SD + LWD: better results compared with LE or LD
  - In the best 5% (and 10%) sequences
    - Higher proportion of LWD at early stage
  - No obvious patterns in the worst LWD hybridizations

- Adaptive heuristic hybridization
  - GHH: focuses on early sequences
  - Adaptively adjust LWD hybridisation

For iterations
- hybridize \( a \%) of LWD into the first half of \( h \)
- produce a solution \( s \) using \( h \)
  - If \( s \) is better or infeasible, increase \( a \)
  - otherwise decrease \( a \)
- Keep the best \( h \) so far
Extension II: Case Based GHH

- Extract/record **knowledge** of heuristic selection during problem solving
- Learn to select and suggest **good heuristics** for particular situations
- Obtained good results on simulated problems, and test on real-world problems
- Assumption: similar problems similar solutions

R. Qu, co-authors: E. Burke, S. Petrovic, **Case Based Heuristic Selection for Timetabling Problems.** Journal of Scheduling, 9: 115-132, 2006. **Top 1% cited by ISI.**
Extension II: Case Based GHH

CBR System

Heuristic Selector

Case Base

Construct Solution

Stop?

problem

solution

NATCEN – Heuristics and Approximate Algorithms, Nottingham, April 2016
Extension II: Case Based GHH

- CBR: suggests good heuristics that worked well in previous similar situations employing knowledge stored in the system
- Case base
  - problems and their partial solutions during problem solving
  - best heuristics for that situations
- Similarity measure: nearest neighbourhood approach
- Key issue of meaningful comparison between two problem solving situations
  - features describe the characteristics of problem and partial solution (cases)
Extension II: Case Based GHH

- Analysis on all possible features
- Training process on feature list
  - Search for most relevant features by which cases (problems and problem solving situations) can be compared concerning most appropriate heuristics used
  - Tabu search
- Training process on cases in case base
  - Leave-one-out strategy: refine the cases stored in case base for problem solving
  - Only cases that may make contribution to problem solving are retained
Extension II: Case Based GHH

Observations

- the more features, the better?
- features selected are more important than their weights in the similarity measure
- search methods for the feature list are not crucial

- vs. graph based hyper-heuristics
- not an easy task for selecting the best meta-heuristics to solve the whole problem
Extension III: GHH Landscape

- Landscape of high level heuristic space
  - More likely to have “walks” or plateau
    - Not mapped to all solutions in solution space (hypothesis)
    - Size of neighbourhoods is very large
    - Computational time: limited number of evaluations within a limited time
  - 1-flip on a heuristic list
  - Fitness distance correlation (fdc): local optimal vs. best
Extension III: GHH Landscape

- Landscape of high level heuristic space
Other Extensions

- Landscape of high level heuristic space
  - More likely to have plateau (neutral)

- Synchronise the search in two search spaces
  - Difficulty of landscape analysis in solution space

- Other recent extensions in the literature
  - Hierarchical hybridisation of graph heuristics
  - ...