***Case-Study HFT-Stuttgart:*** Master Software-Technology – summer term 2010

Repository: <http://code.google.com/p/timetablinghft>

**Heuristic of an Evolutionary Algorithm to solve the curriculum based course timetabling problem**

Frank Borchardt – Steffen Kremer – Roy Pottukalam – Matthias Ruszala –

Alexander Weickmann – Sotiris Zdragkas

**Idea of the heuristic:**

**Generator:**

Fills up the solution table with feasible solutions generated by the *algorithm of Martin Josef Geiger*[[1]](#footnote-1).

**Solution table**

Concrete Solution

Vote

Solution 1

100

Solution 2

10

…

Solution n

20

Rates

**Evaluator**

Fills up

**Generator**

Writes back

Reads

**“Genetist”**

The generator also brings in new genes into the population during the evolution-process.

**Evaluator:**

Evaluates each solution by one concrete curriculum. Based on this rating the reproduction-strategy will take place. The higher the rating is, the higher is the possibility for reproduction of the concrete solution.

**“Genetist”:**

This module reads several solutions from the solution table and creates new (mostly better) solutions via

* Recombination
* Mutation

This module uses the *Neighborhood Analysis by Zhipeng Lü, Jin-Kao Hao and Fred Glover*[[2]](#footnote-2).

**Voting-mechanism in detail – Negotiation approach:**

3

1

9

1

5

6

1

2

3

…

Solution #

Rank

**1**

**2**

3

**Curr1**

**Curr2**

**Curr3**

The evaluator generates one *“negotiator”* (Curr1, Curr2, …) for each curriculum. Each *“negotiator”* votes for every solution in its own interest.

At the end of the negotiation process the highest rated solutions will be used for recombination and mutation.

**Activity Diagram:**

**Definition of fairness:**

For each solution you can calculate the soft constrains for every curriculum.

|  |  |  |  |
| --- | --- | --- | --- |
| **Soft constr. Curr1** | **Soft constr. Curr2** | **Soft constr. Curr3** | **Fairness** |
| 200 | 200 | 200 | 0 |
| 500 | 50 | 50 | 300 |

The fairness points are calculated based on the absolute difference of maximum or minimum penalty (whichever creates higher difference) and the average difference to the other curricula.

The lower the fairness points are, the fairer is the solution.

Not only the highest rated solutions, but also the fairest solutions have a higher possibility for reproduction.

Adequate solution found

Genetist: Read m solutions based on their ratings.

Genetist: Recombination & Mutation.

Genetist: Eliminate the worst k solutions.

Find a “good” solution

Generator: Fill up the solution table until n feasible solutions.

Evaluator: Give a rating for every solution.

**Measurements:**

To represent the quality of our heuristic, here are some significant results of the algorithm. A detailed list with multiple runs on each instance can be found in the repository (address on top of page1). Significant values are marked. In the table of measurements you can see, that (sadly) fair solutions do not have a low penalty value :-(

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Instance | # of Iterations | Best solution penalty | Fairness of best solution | Fairest solution penalty | Faireness of fairest solution |
| 1 | 100 | 1301 | 446 | 1998 | 384 |
| 1 | 1000 | 1077 | 671 | 3256 | 75 |
| 8 | 100 | 8560 | 149 | 11308 | 113 |
| 8 | 1000 | 7388 | 2279 | 8829 | 533 |
| 11 | 100 | 1828 | 663 | 4194 | 6 |
| 11 | 1000 | 1448 | 297 | 2288 | 10 |

**Summary:**

Based on the negotiation approach, not the best solutions will be used for reproduction, but the fairest solutions.

All in all the Algorithm will approximate to the fairest all in all solution.

1. <http://w1.cirrelt.ca/~patat2008/PATAT_7_PROCEEDINGS/Papers/Geiger-TC1d.pdf> [↑](#footnote-ref-1)
2. <http://www.info.univ-angers.fr/pub/hao/papers/JoH2010.pdf> [↑](#footnote-ref-2)