Clustering Heuristic for the Event Timetabling Problem

Peter Karich

University of Bayreuth, Institute of Physics and BIMF
95440 Bayreuth, Germany
peter.karich@uni-bayreuth.de
http://www.timefinder.de

Abstract. We introduce the clustering method based on the "No Collision Principle", which avoids all hard constraint violations by construction.

Key words: Timetabling, Clustering, No Collision Principle, Dynamic Simulated Annealing and Heating, Incremental Assignment Algorithm

Description

The goal of the "No Collision Principle" is to solve the timetabling problems of track 2 specified in [1]. Some additional definitions has to be done:
A "cluster" c_j is a group of events linked to a timeslot t_j. The algorithm avoids hard contraint violations (collisions) within every cluster by construction. One week will be described by the "cluster-array" $\{c_j |\ c = 1..45\}$.
A "raster-collision" occurs if an event is assigned to one of its forbidden timeslots. The "No Collision Principle" [6] works as follows:

Step 1 Put n events into a starting-array A randomly.

Step 2 Correct the order of the events in A with a bubblesort-alike algorithm to prevent order collisions where we have to assume that the events have a cycle free order-graph.

Step 3 For every event e_i in A get all available timeslots: $\{t_j |\ j = m..n,\ m \geq 1,\ m \leq n,\ n \leq 45\}$. They will be used in the next step. Some timeslots could be unavailable because of a predefined raster and some timeslots are unavailable, because of a predefined order. This order is calculated only from the events which are already in the cluster array.

Step 4 For every calculated timeslot t_j of event e_i try to insert the event into the cluster c_j, which is empty at the beginning. If it is not possible to assign[1] a room to the event or if there are person collisions, the algorithm tries the next timeslot. Otherwise the algorithm adds the event to the cluster, assigns the room to the event and proceeds with the next event e_{i+1}, i.e. it continues with step 3. It goes to step 5, if no such event exists.

[1] with the Hungarian Algorithm described e.g. in [2] or with a faster approach [3]

Step 5 Try to add the unplaced events into the cluster array where the algorithm tries to change the start time of every colliding event recursively until a maximal depth (e.g. 4) is reached. If this moving fails, the event stays in the unplaced array.

Step 6 Calculate the percentage p with $p > 0, p < 1$ which will be used in step 8. p depends on the previous iteration(s) and on the current solutions‘ hard and soft constraints[2]. This calculation based on a highly dynamic “Simulated Annealing and Heating” to avoid being trapped in local minima.

Step 7 Choose the current solution as the new local optimum, if it has less unplaced events u than the best solution determined so far or - if u is zero - compare the soft constraints. Stop if there are no hard and soft constraints.

Step 8 Randomly put $p \cdot n$ events from the cluster array back to the starting-array A. Go to step 2, if we have enough time otherwise stop.

Summary

The “No Collision Principle” was applied successfully to the given datasets [1]. All hard constraint violations could be removed and a mean value of the soft constraint violations under 1600 ± 600 was reached for the provided 16 datasets within the limited time.

Acknowledgment

I would like to thank the following people: My wife and my son for their great help all the time; Peter Wilke (University of Erlangen-Nuremberg) for his valuable discussions; My parents - not only for their financial support; Lothar Kador (University of Bayreuth) for his support and ideas in the lab; The Java and NetBeans community for its great products and YourKit for its profiler.

References

1. The Second International Timetabling Competition 2007/2008 `http://www.cs.qub.ac.uk/eventmap/`
2. Clark, J., Holton, D.A.: Graphentheorie - Grundlagen und Anwendungen, Spektrum - Akademischer Verlag (1994)
3. Toroslua, I.H.: Incremental Assignment Algorithm (2005), Information Sciences, Volume 177, Issue 6, 15 March 2007, Pages 1523-1529
4. The solver is available under `http://www.timefinder.de` or `http://gstpl.sourceforge.net`
5. An example will be available under `http://gstpl.wikispaces.com/International+Timetabling+Competition`
6. No Collision Principle, `http://gstpl.svn.sourceforge.net/viewvc/gstpl/trunk/main/src/de/gstpl/algo/localsearch/NoCollisionPrincipleA.java?view=markup`

[2] Starting from $p = 0.2$