KHE14: An Algorithm for High School Timetabling

Jeffrey H. Kingston

The University of Sydney
KHE14

- Solves high school timetabling problems given in XHSTT format
- Built on the KHE platform
- Aims to solve a wide range of instances (XHSTT-2014) quickly and well
The XHSTT format (sketch)

- An XML format
- Times, resources (teachers, rooms, classes, etc.), events, and constraints
- Sets of times, resources, events
- 16 constraint types: spread through the week, unavailable times, etc.

Badness of solution = weighted sum of constraint violations
The parts of KHE14

- Structures (for representing courses, linked events, etc.)
- Global tixel matching (monitors availability of resources)
- Time and resource assignment and repair phases

Most of the new work is in the repair phases.
Repair using ejection chains

For each defect (each point where a constraint is violated):

- Repair the defect, possibly introducing new defects
- If no new defects, success; restart on next defect
- If two or more new defects, undo and try next repair
- If one new defect, recurse and so build chain of repairs

Two ways to stop: limit chain depth; mark changed things tabu.

Key point: a new defect may be completely new, with no history of failure to repair.
Polymorphic ejection chains

- Each defect may have a different type
- Each repair may be tailored to the type of defect
Repair operations

- Unassign, assign (basic or ejecting), move (basic or ejecting), swap, Kempe-move
- Applicable to both times and resources
- Event splitting and merging, combined with Kempe move
- Sets of these

Each defect type has its own specific set of repairs.
Example 1: repairing limit busy times underloads

A resource $r$ is underloaded during some set of times $T$.

- Move one event containing $r$ so that its overlap with $T$ increases
- Move all events overlapping $T$ away from $T$

Important lesson: more precisely targeted repairs give better results.
Example 2: repairing cluster busy times overloads

A resource $r$ is busy on too many days $D$. For each day $d$ in $D$:

- Unassign $r$’s events overlapping $d$
- Reduce the domains of all $r$’s events to exclude all empty days
- Continue with an ejection tree rather than ejection chain
- Restore original domains afterwards, whether succeed or not

Notes:
- Cluster overloads are difficult: repair usually requires moving several events
- Only do this on main-loop defects; too expensive in general
Many more details in (revised) printed paper

• Results on XHSTT-2014: several new bests; repair slower than ideal
• Refinements to the assignment phases which reduce defects
• Tailored repairs for 16 kinds of defects
• Experiments comparing variants of the basic ideas