KHE24: Towards a Practical Solver for Nurse Rostering

Jeffrey H. Kingston (University of Sydney)
http://jeffreykingston.id.au

This talk

- The nurse rostering problem
- The KHE24 solver
- Success in practice
- Results
- Interesting ways to improve timetables

The nurse rostering problem

- Assign nurses to shifts in hospital wards
- 24 / 7 operation, so morning, afternoon, night etc. shifts
- *Cover constraints* for shifts: 3 to 5 nurses, at least 1 senior nurse, …
- *Resource constraints* for nurses:
  - At most one shift per day
  - Counters: at most 20 busy days; at most 2 busy weekends, …
  - Sequences: at most 5 consecutive night shifts, …
  - Unwanted patterns: day shift after night shift, incomplete weekends, …

About the KHE24 nurse rostering solver

- Built on the KHE solve platform (read, write, evaluate solutions, etc.)
- Aims to find a good but not optimal solution quickly and reliably
- Runs in polynomial time (weighted bipartite matching, ejection chains)
- Uses the XESTT data format, which is good because:
  - Many well-known instances are available in XESTT
  - XESTT uses only 9 constraint types (3 cover, 6 resource)
- Work in progress

The solver's algorithm (simplified)

- Find initial solution using time sweep (PATAT 2018)
- Repair using
  - Polymorphic ejection chains (PATAT 2012)
  - Optimal reassignment by dynamic programming (PATAT 2022). Also called VLSN search, ruin and replace, matheuristics (when IP solver used), etc., etc.
- Run 12 independent solves in parallel, keep the best solution. Integral part of solver, not just for testing.

Success in practice

A solver is **successful in practice** if, on every instance likely to occur in practice:

- It finds a solution with cost within 10% of the best known within 5 minutes
- And it finds a solution with cost within 5% of the best known within 60 minutes

Points to note:

- An alternative to finding new best solutions, not a replacement.
- Implies testing on a wide range of instances.
- Challenging despite not requiring new best solutions.

Results

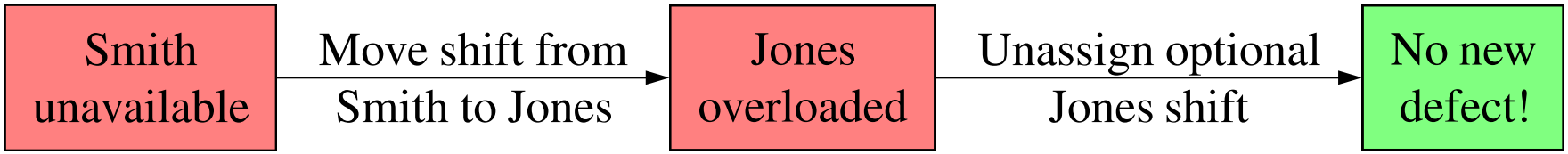
Results of 5-minute and 60-minute runs on 107 instances in paper:

- Curtois original instances – mostly good, some problems with the larger instances
- First International Timetabling Competition – just a few problem instances
- Second International Timetabling Competition – not yet good enough
- Curtois and Qu 2014 instances – not tested yet

The work continues.

Ejection chains

- Visit each *defect* (constraint violation) and try to repair it
- One repair can create another defect, leading to a chain of repairs:



- No new defects means success; 2 or more new defects means failure
- Repairs are *polymorphic*: they depend on the type of defect
- … but most swap two nurses' timetables over a sequence of consecutive days
- Stop at time limit, or when all defects fail to repair

Ejection chains: Nurse is busy during unavailable time

	Night	Night	Night			Day	Day		
				Night	Night	Night	Night	Night	

				Night	Night	Night	Night	Night	
	Night	Night	Night			Day	Day		

!

Polymorphism: try whole-timetable swap to fix unavailable times defects **only**.

Ejection chains: Night shifts must occur in sequences of at least 4

	Night	Night	Night			Day	Day		
				Night	Night	Night	Night	Night	

over	Night	Night	Night	Night		Day	Day		
					Night	Night	Night	Night	

• • •

Ejection chains: Double swaps

	Night	Night	Night			Day	Day		
				Night	Night	Night	Night	Night	
over				Day	Day	Day	Day	Day	

over	Night	Night	Night	Night		Day	Day		
				Day	Day	Day	Day	Day	
					Night	Night	Night	Night	

!