KHE18: A Solver for Nurse Rostering

Jeffrey H. Kingston (University of Sydney)

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 least 2 free weekends, …
  - Sequences: at most 4 consecutive busy days, 2 consecutive weekends, …
  - Unwanted patterns: day shift after night shift, incomplete weekends, …

The KHE18 nurse rostering solver

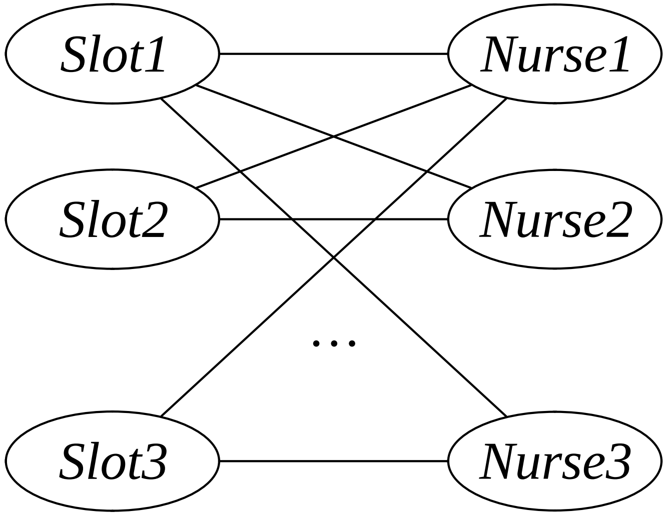
- Built on the KHE solve platform
- Descended from the KHE14 high school timetabling solver
- Aims to find a good but not optimal solution quickly and reliably
- Runs in polynomial time (weighted bipartite matching, ejection chains)
- KHE18x8: run KHE18 in parallel 8 times, keep best solution
- Work in progress

The XESTT nurse rostering data format

- A new format (see two other PATAT 2018 papers)
- KHE18 reads instances in XESTT format
- XESTT has two advantages:
  - Many well-known instances are available in XESTT
  - XESTT uses only 9 constraint types, so constraint-specific code is feasible

Initial assignment using time sweep

- Assign nurses to the shifts of day 1, then day 2, etc.
- Use weighted bipartite matching for assigning each day:



This assumes that each nurse can take at most one shift per day.

Ideas for improving time sweep

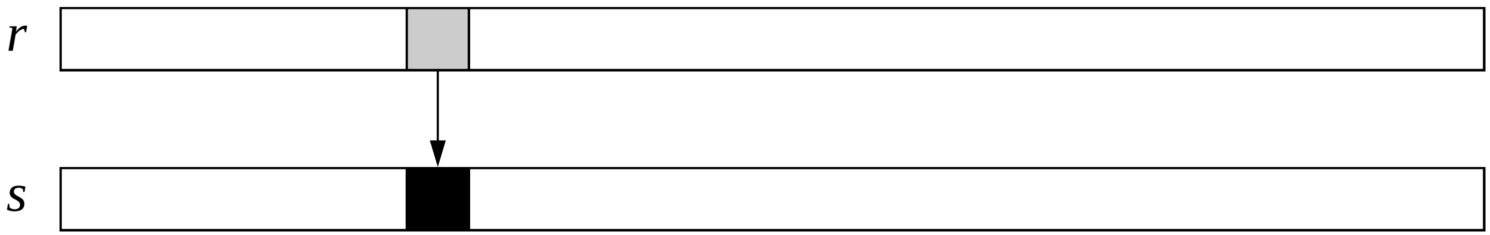
- Repair as you go (unassign and reassign one recent day)
- Suppress spurious costs arising from minimum limits
- Break edge weight ties to favour underutilized resources
- Make special arrangements for limit resources constraints

Repair using ejection chains

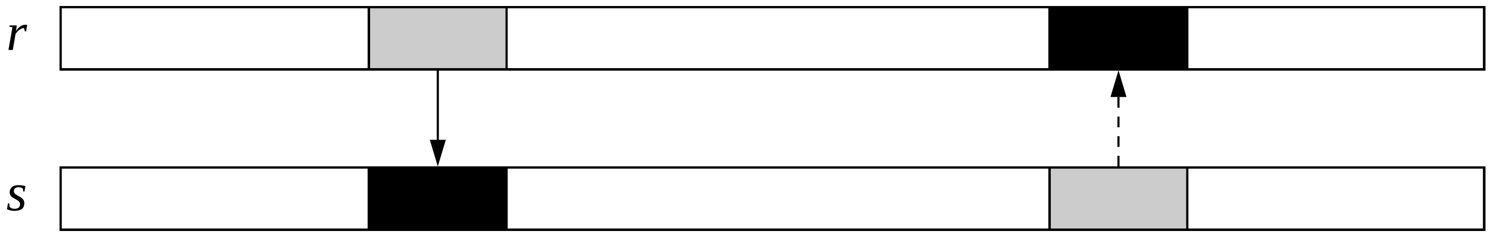
- Visit each *defect* (constraint violation) and try to repair it
- One repair can create another defect, leading to a chain of repairs
- No new defects means success; 2 or more new defects means failure
- Stop at time limit, or when all defects fail to repair
- Repairs are *polymorphic*: depend on the type of defect

Example: nurse *r* overloaded during week *w*

- Fundamental repairs: all moves of a week *w* shift assigned *r* to another resource *s*:



- But in nurse rostering it is usually better to move several adjacent days:



- And to optionally move a different run (same length) from *s* to *r*

Conclusion

- Results in paper for KHE18 and KHE18x8 on three well-known data sets
- Running times generally good (e.g. 41 seconds max per instance on INRC1)
- Costs good sometimes (e.g. next version optimal on 6 COI instances)
- But poor sometimes (e.g. feasible for only 7 of 24 CQ14 instances)
- As always, more work is needed