Improving the Dynamic Programming Algorithm for Nurse Rostering

Jeffrey H. Kingston

The University of Sydney

Overview

Existing algorithm

- A dynamic programming algorithm for optimally timetabling one nurse
- Found in column generation algorithms for nurse rostering
- Runs in time polynomial in the number of days to be timetabled

This paper's contribution

- Generalize to multiple nurses, any subset of the days, all constraints
- Speed up (essential for multiple nurses)

Aim is to use it as reconstruction operator in VLSN search (still to do).

Nurse rostering

Assign shifts (e.g. morning, afternoon, night) to the nurses of a hospital ward, over several weeks.

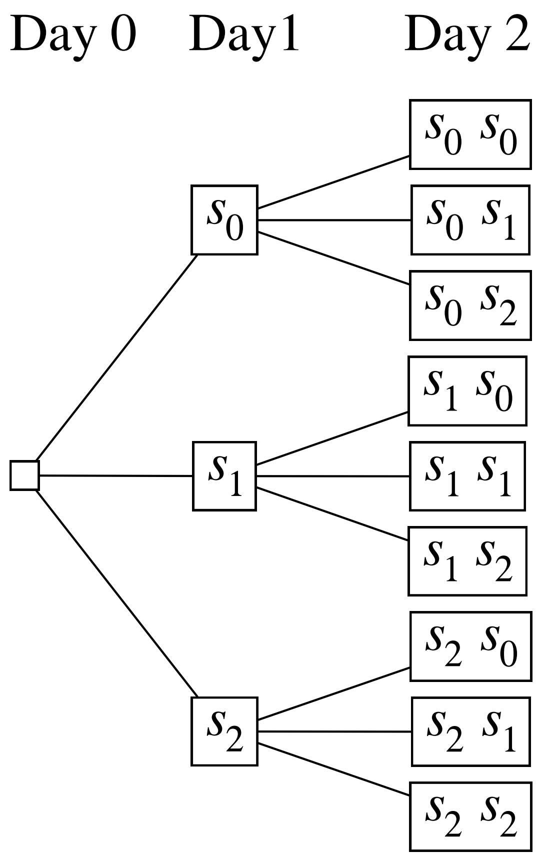
Cover constraints

- For each shift, limits on the number of nurses assigned
- Requests for nurses may specify skills (senior, trainee, etc.)

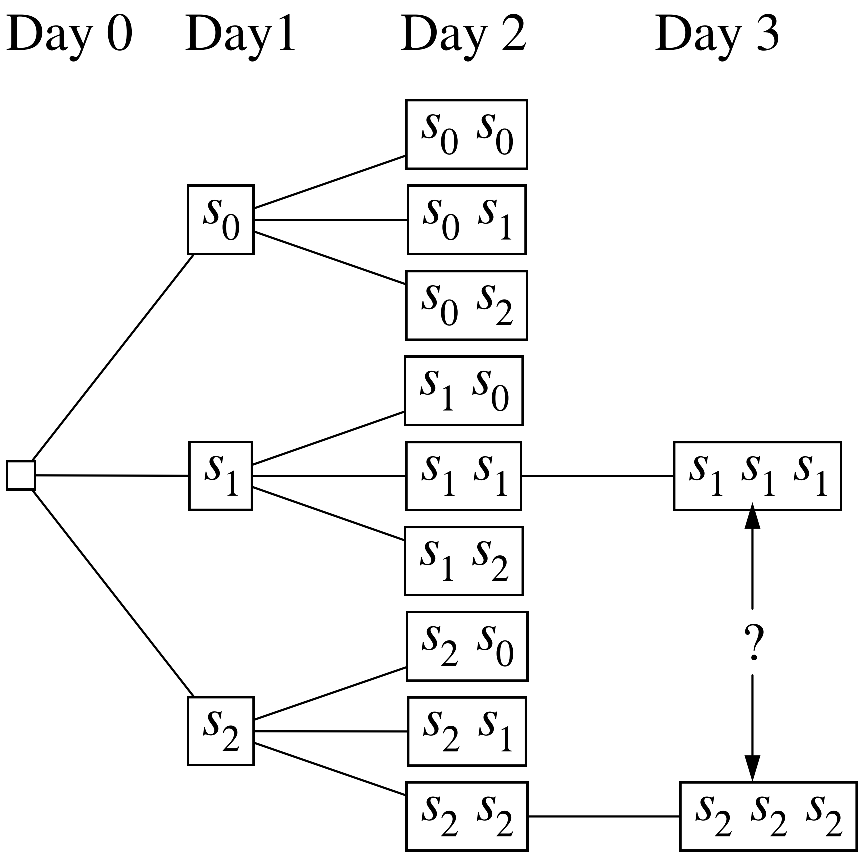
Resource constraints

- Each nurse takes at most one shift per day (hard)
- Counter constraints: limits on total shifts, busy weekends, etc.
- Sequence constraints: limits on consecutive busy days, night shifts, etc.

Simple tree search (single nurse)



The dynamic programming algorithm



Dominance testing

Complete extension of solution S

A solution that begins with S and carries on to the end.

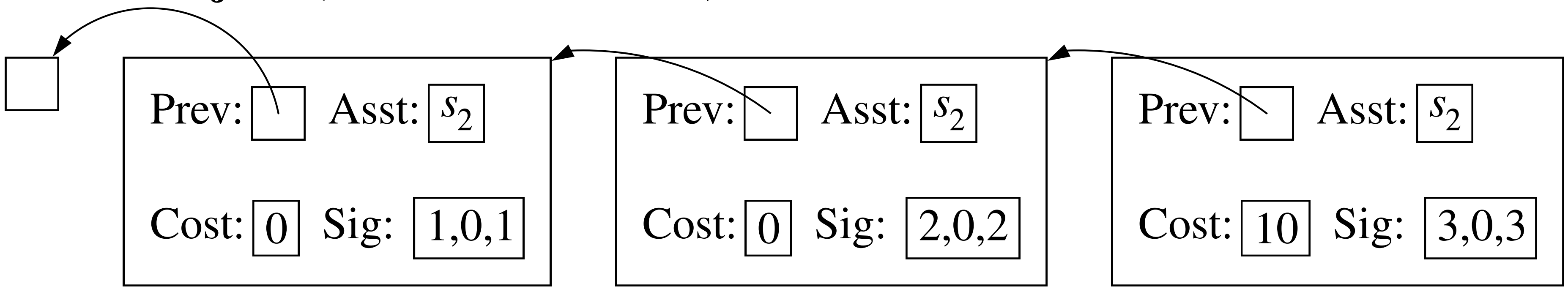
Dominates(S_1, S_2)

True when for each complete extension of S_2 there is a complete extension of S_1 of equal or less cost.

If Dominates(S_1, S_2) then we can delete S_2. Check this as each solution is created.

How solutions are represented

Solution objects (nodes of search tree)



Signatures

Constraint	$\boxed{s_1\ s_1\ s_1}$	$\boxed{s_2\ s_2\ s_2}$
At most 5 shifts	3	3
At most 3 consecutive s_1 shifts	3	0
At most 2 consecutive s_2 shifts	0	3 + cost 10 (say)

Basic dominance

S_1 dominates S_2 when

- $cost(S_1) \le cost(S_2)$ and
- for each constraint m in the signatures,
  - if upper limit only, $sig(m, S_1) \le sig(m, S_2)$
  - if lower limit only, $sig(m, S_1) \ge sig(m, S_2)$
  - if both, $sig(m, S_1) = sig(m, S_2)$

Seems to be the standard in the literature.

Tradeoff dominance

- Seems to be new to this paper
- Similar to basic dominance
- But can trade off a small violation against $cost(S_2) - cost(S_1)$

Uniform dominance

- Done since paper was written
- Similar to tradeoff dominance
- Handles awkward cases well (allow zero flag, quadratic cost functions)
- May be best possible dominance test in practice

Generalizing

Multiple nurses

One assignment for each nurse, and all nurses' constraints in the signature:

Prev: $\boxed{}$ Asst: $\boxed{r_1 := s_2, r_2 := s_0}$

Cost: $\boxed{0}$ Sig: $\boxed{2,0,2,2,2,0}$

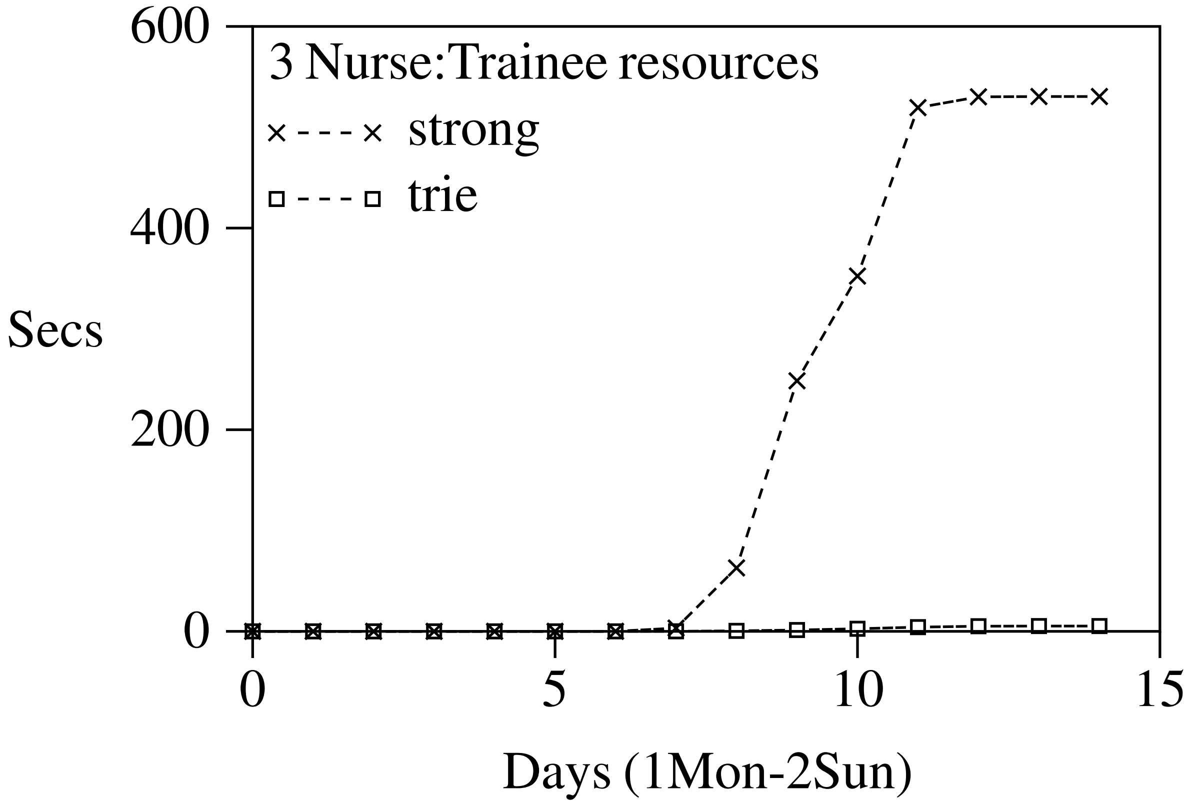
Any subset of the days

Search over selected days only; unselected days add constants to signatures.

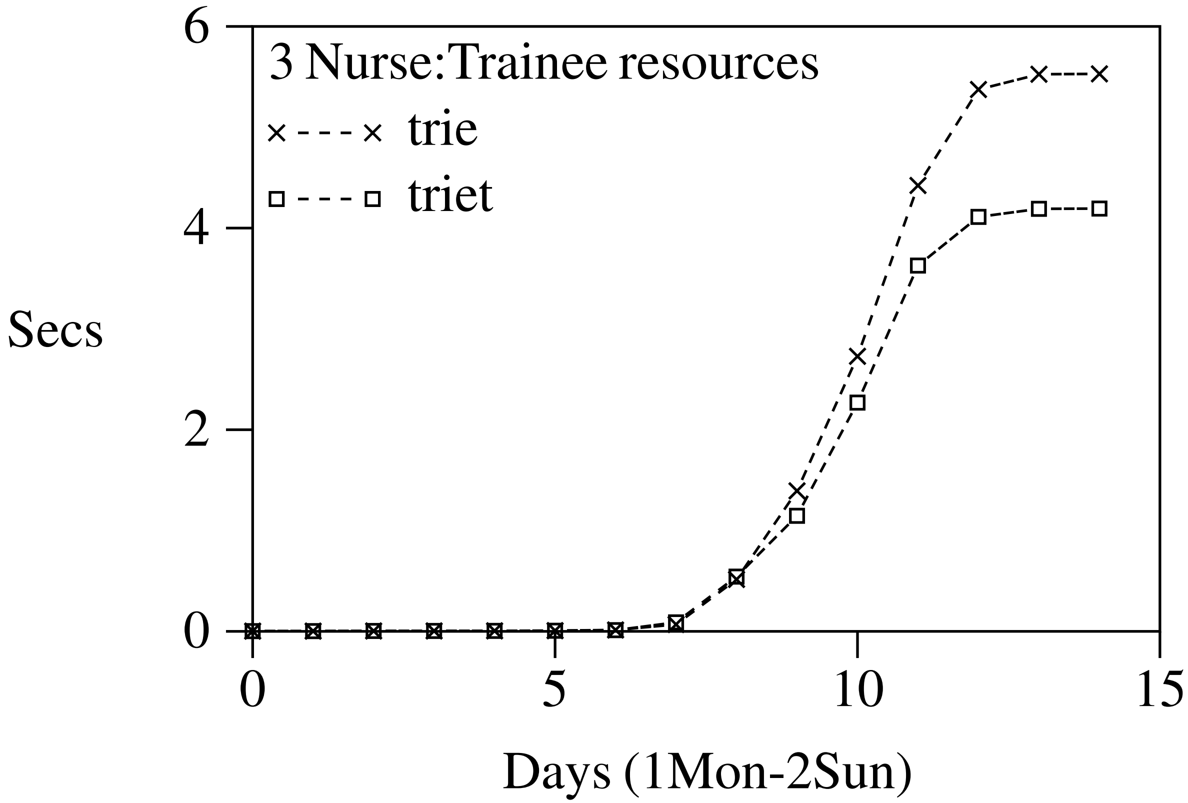
All constraints

Support the XESTT format; its constraints cover all the well-known data sets.

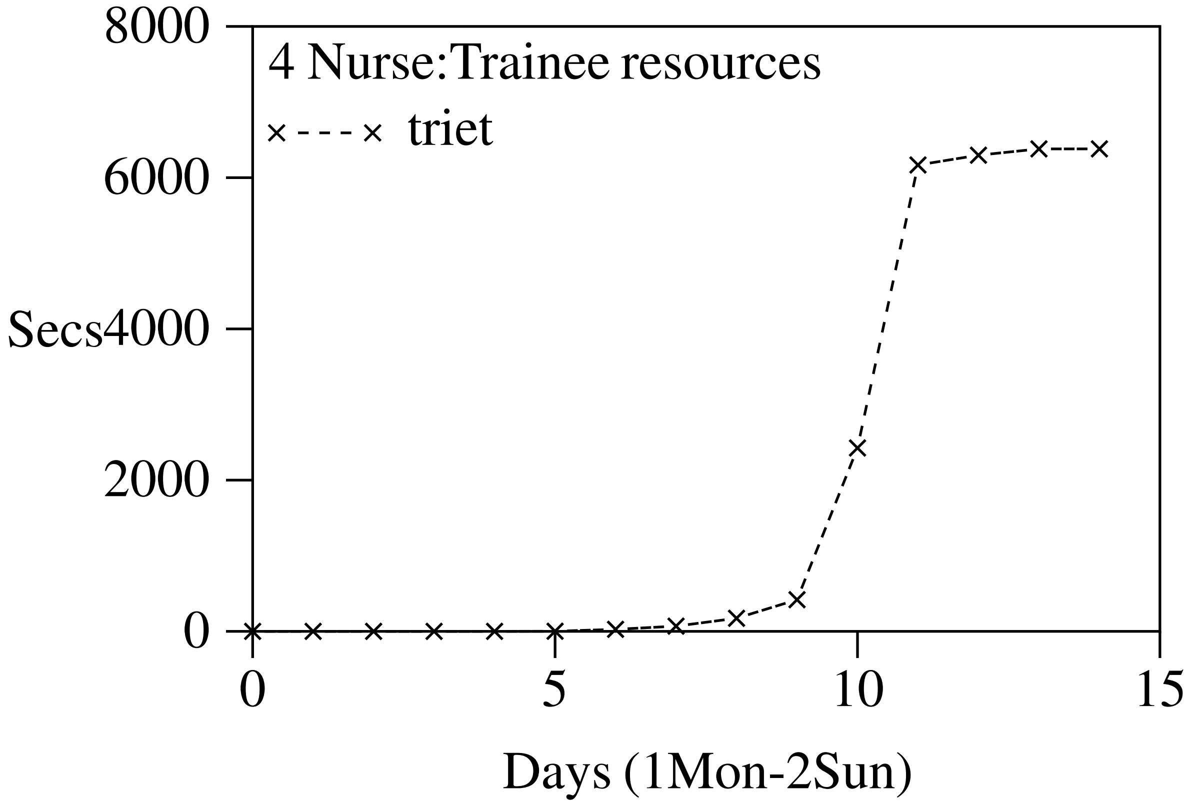
Experiment 1 – Storing solutions in a trie data structure



Experiment 2 – Moving to tradeoff dominance



Experiment 3 – Four nurses not yet efficient



Conclusion

What has been done

- Algorithm generalized to arbitrary nurses, days, constraints
- And made to run much faster

What still needs to be done

- More speedup (5 nurses at least)
- Test algorithm as VLSN reconstruction operator