Presentation at CP 2011:

A hybrid approach for solving real-world nurse rostering problems

Martin Stølevik (martin.stolevik@sintef.no)
Tomas Eric Nordlander (tomas.nordlander@sintef.no)
Atle Riise (atle.riise@sintef.no)
Helle Frøyseth (hellef@gmail.com)

Content

- The Nurse Rostering Problem
  - Model
  - Constraints
- The Hybrid Solution method
  - Constraint Programming
  - Variable Neighbourhood Descent
  - Focal Points
  - Destruction of Rosters
- Results
  - The Developed Software

The nurse rostering problem

The nurse rostering problem (NRP)

- Producing good nurse rosters is a complex and important task.
- The demand for personnel varies over time.
- The rosters must obey
  - Labour laws
  - Union regulations
  - Hospital policies
- There are multiple stakeholders to satisfy, with, often, conflicting objectives:
  - Employees
  - Employer
  - Patients

NRP model

- Nurse rostering:
  To allocate shifts to nurses over the scheduling period while satisfying hard constraints and minimizing violations to soft constraints. (…so typically an over-constrained optimization problem)
- Shifts
  - Duration
  - Competence / Groups
- Shift categories (D, E , N)
- Employees
  - Working time / Contracts
  - Wishes
  - Competence / Group
- Days

Employee	0	1	2	3	4	5	6	7	8	9	10	11	12	13
399542381	<off>	D	D	E1	D3	D3	D	<off>	<off>	N	N	N	<off>	<off>
399582845	N	E	D3	D	<off>	<off>	<off>	D3	D8	<off>	<off>	<off>	D	D
399563055	<off>	E2	E1	E1	D	E1	E1	<off>	<off>	N	N	<off>	<off>	<off>
399568044	<off>	<off>	D	D	DH	<off>	<off>	D3	D	<off>	<off>	D3	E1	E1
399545140	N	N	E1	<off>	<off>	E1	E1	<off>	<off>	D	D3	D	<off>	<off>
399519405	<off>	<off>	<off>	<off>	<off>	<off>	<off>	D	E1	<off>	<off>	<off>	<off>	<off>
399510055	D	D	D	E1	D	<off>	<off>	N	N	E1	DH	D	<off>	<off>
399577605	D	D8	E1	DH	D8	<off>	<off>	D	D	<off>	<off>	N	N	N
399532310	D3	N	N	<off>	<off>	D	D	D	E1	D	E1	<off>	<off>	<off>
399576231	E1	D3	<off>	<off>	E1	D	D	<off>	<off>	D3	D	E1	<off>	<off>
399504383	D	D	D	E1	E1	<off>	<off>	D	E1	E	<off>	<off>	E1	E1
399522121	E1	D	<off>	<off>	<off>	<off>	<off>	<off>	<off>	E1	D	<off>	<off>	<off>
399500880	<off>	E1	D3	D8	D8	<off>	<off>	N	N	E1	<off>	<off>	E1	E1
399544861	<off>	<off>	<off>	E1	D	D	D	<off>	<off>	E1	N	N	<off>	<off>
399530839	D3	D	<off>	<off>	<off>	D3	D	N	N	N	N	<off>	<off>	<off>
399590468	D	E	E1	D	<off>	<off>	<off>	D8	E1	E	D3	D3	<off>	<off>

Hard constraints

- Match the cover specified per day exactly
- Working hours (over the scheduling period) must be with in given limits
- Match competence requirement on shifts to nurses
- Minimum resting time between two shifts
- Weekly free period of minimum duration
- Maximum weekly working time
- (Assign one shift per day per nurse)

Employee	0	1	2	3	4	5	6	7	8	9	10	11	12	13
399542381	<off>	D	D	E1	D3	D3	D	<off>	<off>	N	N	N	<off>	<off>
399582845	N	E	D3	D	<off>	<off>	<off>	D3	D8	<off>	<off>	<off>	D	D
399563055	<off>	E2	E1	E1	D	E1	E1	<off>	<off>	N	N	<off>	<off>	<off>
399568044	<off>	<off>	D	D	DH	<off>	<off>	D3	D	<off>	<off>	D3	E1	E1
399545140	N	N	E1	<off>	<off>	E1	E1	<off>	<off>	D	D3	D	<off>	<off>
399519405	<off>	<off>	<off>	<off>	<off>	<off>	<off>	D	E1	<off>	<off>	<off>	<off>	<off>
399510055	D	D	D	E1	D	<off>	<off>	N	N	E1	DH	D	<off>	<off>
399577605	D	D8	E1	DH	D8	<off>	<off>	D	D	<off>	<off>	N	N	N
399532310	D3	N	N	<off>	<off>	D	D	D	E1	D	E1	<off>	<off>	<off>
399576231	E1	D3	<off>	<off>	E1	D	D	<off>	<off>	D3	D	E1	<off>	<off>
399504383	D	D	D	E1	E1	<off>	<off>	D	E1	E	<off>	<off>	E1	E1
399522121	E1	D	<off>	<off>	<off>	<off>	<off>	<off>	<off>	E1	D	<off>	<off>	<off>
399500880	<off>	E1	D3	D8	D8	<off>	<off>	N	N	E1	<off>	<off>	E1	E1
399544861	<off>	<off>	<off>	E1	D	D	D	<off>	<off>	E1	N	N	<off>	<off>
399530839	D3	D	<off>	<off>	<off>	D3	D	N	N	N	N	<off>	<off>	<off>
399590468	D	E	E1	D	<off>	<off>	<off>	D8	E1	E	D3	D3	<off>	<off>

Soft constraints

- Max and min number of consecutive working days
- Max and min number of consecutive days of same shift category
- Max and min number of shifts
- Max and min number of shifts in shift categories
- Minimize deviation to contracted working time
- Cluster days off
- Maximise the number of wanted patterns
- Minimise the number of unwanted patterns

Employee	0	1	2	3	4	5	6	7	8	9	10	11	12	13
399542381	<off>	D	D	E1	D3	D3	D	<off>	<off>	N	N	N	<off>	<off>
399582845	N	E	D3	D	<off>	<off>	<off>	D3	D8	<off>	<off>	<off>	D	D
399563055	<off>	E2	E1	E1	D	E1	E1	<off>	<off>	N	N	<off>	<off>	<off>
399568044	<off>	<off>	D	D	DH	<off>	<off>	D3	D	<off>	<off>	D3	E1	E1
399545140	N	N	E1	<off>	<off>	E1	E1	<off>	<off>	D	D3	D	<off>	<off>
399519405	<off>	<off>	<off>	<off>	<off>	<off>	<off>	D	E1	<off>	<off>	<off>	<off>	<off>
399510055	D	D	D	E1	D	<off>	<off>	N	N	E1	DH	D	<off>	<off>
399577605	D	D8	E1	DH	D8	<off>	<off>	D	D	<off>	<off>	N	N	N
399532310	D3	N	N	<off>	<off>	D	D	D	E1	D	E1	<off>	<off>	<off>
399576231	E1	D3	<off>	<off>	E1	D	D	<off>	<off>	D3	D	E1	<off>	<off>
399504383	D	D	D	E1	E1	<off>	<off>	D	E1	E	<off>	<off>	E1	E1
399522121	E1	D	<off>	<off>	<off>	<off>	<off>	<off>	<off>	E1	D	<off>	<off>	<off>
399500880	<off>	E1	D3	D8	D8	<off>	<off>	N	N	E1	<off>	<off>	E1	E1
399544861	<off>	<off>	<off>	E1	D	D	D	<off>	<off>	E1	N	N	<off>	<off>
399530839	D3	D	<off>	<off>	<off>	D3	D	N	N	N	N	<off>	<off>	<off>
399590468	D	E	E1	D	<off>	<off>	<off>	D8	E1	E	D3	D3	<off>	<off>

About constraints

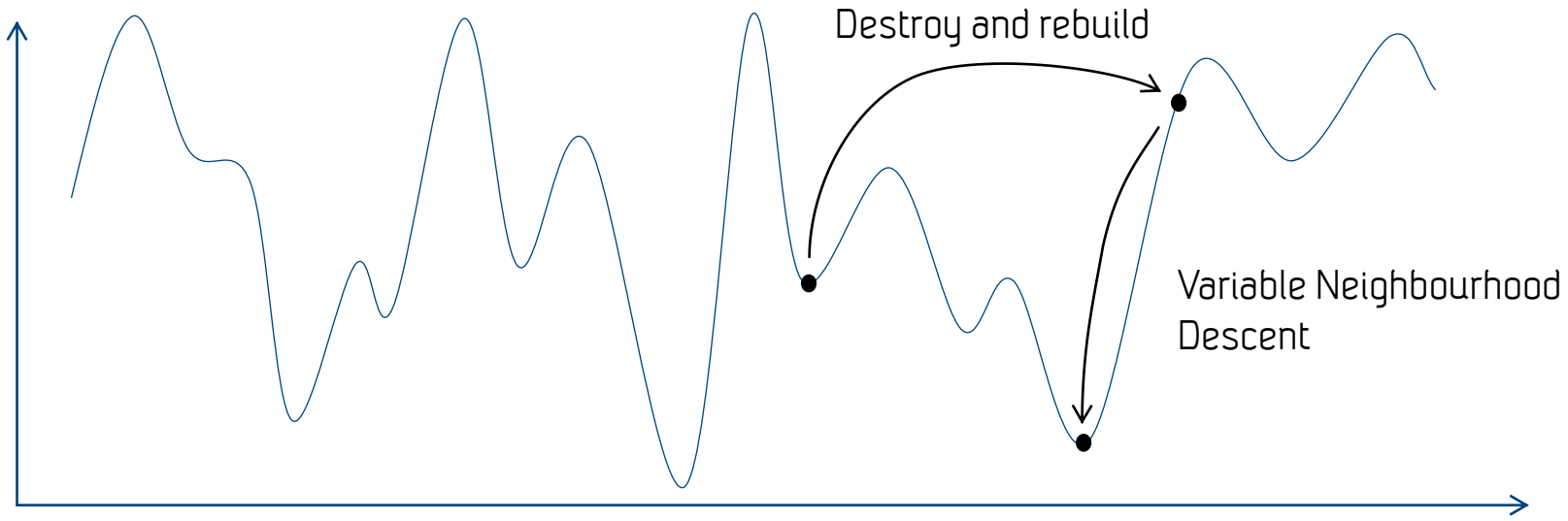
- One vertical constraint; Cover
- One constraint handled implicitly by solution method design; One shift per day
- The rest of the (soft and hard) constraints are "horizontal and per nurse"
- Easy to compute the impact of each roster (nurse) on the overall solution quality

Employee	0	1	2	3	4	5	6	7	8	9	10	11	12	13
399542381	<off>	D	D	E1	D3	D3	D	<off>	<off>	N	N	N	<off>	<off>
399582845	N	E	D3	D	<off>	<off>	<off>	D3	D8	<off>	<off>	<off>	D	D
399563055	<off>	E2	E1	E1	D	E1	E1	<off>	<off>	N	N	<off>	<off>	<off>
399568044	<off>	<off>	D	D	DH	<off>	<off>	D3	D	<off>	<off>	D3	E1	E1
399545140	N	N	E1	<off>	<off>	E1	E1	<off>	<off>	D	D3	D	<off>	<off>
399519405	<off>	<off>	<off>	<off>	<off>	<off>	<off>	D	E1	<off>	<off>	<off>	<off>	<off>
399510055	D	D	D	E1	D	<off>	<off>	N	N	E1	DH	D	<off>	<off>
399577605	D	D8	E1	DH	D8	<off>	<off>	D	D	<off>	<off>	N	N	N
399532310	D3	N	N	<off>	<off>	D	D	D	E1	D	E1	<off>	<off>	<off>
399576231	E1	D3	<off>	<off>	E1	D	D	<off>	<off>	D3	D	E1	<off>	<off>
399504383	D	D	D	E1	E1	<off>	<off>	D	E1	E	<off>	<off>	E1	E1
399522121	E1	D	<off>	<off>	<off>	<off>	<off>	<off>	<off>	E1	D	<off>	<off>	<off>
399500880	<off>	E1	D3	D8	D8	<off>	<off>	N	N	E1	<off>	<off>	E1	E1
399544861	<off>	<off>	<off>	E1	D	D	D	<off>	<off>	E1	N	N	<off>	<off>
399530839	D3	D	<off>	<off>	<off>	D3	D	N	N	N	N	<off>	<off>	<off>
399590468	D	E	E1	D	<off>	<off>	<off>	D8	E1	E	D3	D3	<off>	<off>

Solution method

Iterated Local Search

- The solution method framework is Iterated Local Search (ILS)
  - CP used for initial solution construction (only hard constraints)
  - Iterate between
    - Variable Neighbourhood Descent (VND) and
    - destroy part of the solution and rebuild using CP



ILS with CP hybrid - Pseudo code

```
1  IteratedLocalSearch
2      x* ← x ← CPBuild(x_0)
3      repeat
4          x  ← VariableNeighborhoodDescent(x)
5          x* ← Accept(x, x*)
6          repeat // Diversification
7              x' ← DestroyPartsOfSolution(x)
8              x' ← CPBuild(x')
9          until x' is a legal solution
10         x ← x'
11     until some termination condition is met
12     return x*
```

Constraint Programming

- One variable per day and nurse.
- Domain of the variables: the possible shifts.
- Only need to satisfy the hard constraints
- Aiming for any feasible solution, as fast as possible:

1. First try to solve with all constraints
2. Second with just cover and working time
3. More and more of the hard constraints

Employee	0	1	2	3	4	5	6	7	8	9	10	11	12	13
399542381	<off>	D	D	E1	D3	D3	D	<off>	<off>	N	N	N	<off>	<off>
399582845	N	E	D3	D	<off>	<off>	<off>	D3	D8	<off>	<off>	<off>	D	D
399563055	<off>	E2	E1	E1	D	E1	E1	<off>	<off>	N	N	<off>	<off>	<off>
399568044	<off>	<off>	D	D	DH	<off>	<off>	D3	D	<off>	<off>	D3	E1	E1
399545140	N	N	E1	<off>	<off>	E1	E1	<off>	<off>	D	D3	D	<off>	<off>
399519405	<off>	<off>	<off>	<off>	<off>	<off>	<off>	D	E1	<off>	<off>	<off>	<off>	<off>
399510055	D	D	D	E1	D	<off>	<off>	N	N	E1	DH	D	<off>	<off>
399577605	D	D8	E1	DH	D8	<off>	<off>	D	D	<off>	<off>	N	N	N
399532310	D3	N	N	<off>	<off>	D	D	D	E1	D	E1	<off>	<off>	<off>
399576231	E1	D3	<off>	<off>	E1	D	D	<off>	<off>	D3	D	E1	<off>	<off>
399504383	D	D	D	E1	E1	<off>	<off>	D	E1	E	<off>	<off>	E1	E1
399522121	E1	D	<off>	<off>	<off>	<off>	<off>	<off>	<off>	E1	D	<off>	<off>	<off>
399500880	<off>	E1	D3	D8	D8	<off>	<off>	N	N	E1	<off>	<off>	E1	E1
399544861	<off>	<off>	<off>	E1	D	D	D	<off>	<off>	E1	N	N	<off>	<off>
399530839	D3	D	<off>	<off>	<off>	D3	D	N	N	N	N	<off>	<off>	<off>
399590468	D	E	E1	D	<off>	<off>	<off>	D8	E1	E	D3	D3	<off>	<off>

Constraint programing (2)

- Variables; X_{ed}.
- Handling the competence constraint by initial reduction of the domain of variables.
- Many constraints are handled by expressions. Example: workload.
- Initial propagation (arc consistency).
- During search we maintain arc consistency (MAC).
- Search heuristic: Depth-first search with dynamic variable and value ordering.
  - Select day (partially selecting variable); $X_{e\mathbf{d}}$
  - Select shift category - and choose random shift in that category (value)
  - Select nurse (variable); $X_{\mathbf{ed}}$

Variable Neighbourhood Search

- No moves that violate hard constraints.
- Cover constraint is obeyed by neighbourhood design (only vertical swaps).
- In each iteration, apply first move that decreases penalty.
- Search the neighbourhoods in sequence.
- Need to focus on the "problematic" areas of the current solution.

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Employee 1	E	D	N		E		E
Employee 2			D	E	N		D
Employee 3	D	N		D			
Employee 4	N	E				E	
Employee 5			E		D	N	N

Focal points

- The neighbourhood to search for each of the three local moves quickly become very large
  - Even the smallest one, the simple swap, has the size of $|E|^2|D|$.
  - We must focus on the most promising moves.
- Focal points are features ("places"/variables) in the current solution where changes are likely to yield improvements.
- We create one focal point per variable involved in soft constraint violation

Employee	0	1	2	3	4	5	6	7	8	9	10	11	12	13
399542381	<off>	D	D	E1	D3	D3	D	<off>	<off>	N	N	N	<off>	<off>
399582845	N	E	D3	D	<off>	<off>	<off>	D3	D8	<off>	<off>	<off>	D	D
399563055	<off>	E2	E1	E1	D	E1	E1	<off>	<off>	N	N	<off>	<off>	<off>
399568044	<off>	<off>	D	D	DH	<off>	<off>	D3	D	<off>	<off>	D3	E1	E1
399545140	N	N	E1	<off>	<off>	E1	E1	<off>	<off>	D	D3	D	<off>	<off>
399519405	<off>	<off>	<off>	<off>	<off>	<off>	<off>	D	E1	<off>	<off>	<off>	<off>	<off>
399510055	D	D	D	E1	D	<off>	<off>	N	N	E1	DH	D	<off>	<off>
399577605	D	D8	E1	DH	D8	<off>	<off>	D	D	<off>	<off>	N	N	N
399532310	D3	N	N	<off>	<off>	D	D	D	E1	D	E1	<off>	<off>	<off>
399576231	E1	D3	<off>	<off>	E1	D	D	<off>	<off>	D3	D	E1	<off>	<off>
399504383	D	D	D	E1	E1	<off>	<off>	D	E1	E	<off>	<off>	E1	E1
399522121	E1	D	<off>	<off>	<off>	<off>	<off>	<off>	<off>	E1	D	<off>	<off>	<off>
399500880	<off>	E1	D3	D8	D8	<off>	<off>	N	N	E1	<off>	<off>	E1	E1
399544861	<off>	<off>	<off>	E1	D	D	D	<off>	<off>	E1	N	N	<off>	<off>
399530839	D3	D	<off>	<off>	<off>	D3	D	N	N	N	N	<off>	<off>	<off>
399590468	D	E	E1	D	<off>	<off>	<off>	D8	E1	E	D3	D3	<off>	<off>

Destruction of rosters

- We destroy or unassign, a number of rosters to achieve diversification.
- The destroyed rosters are rebuilt by using CP, keeping the other rosters locked (variables fixed).
- The number of rosters to ruin is picked randomly (between limits) as a mix of
  - the worst rosters (cf. focal points) and
  - some picked at random.

Employee	0	1	2	3	4	5	6	7	8	9	10	11	12	13
399542381	<off>	D	D	E1	D3	D3	D	<off>	<off>	N	N	N	<off>	<off>
399582845														
399563055	<off>	E2	E1	E1	D	E1	E1	<off>	<off>	N	N	<off>	<off>	<off>
399568044	<off>	<off>	D	D	DH	<off>	<off>	D3	D	<off>	<off>	D3	E1	E1
399545140														
399519405	<off>	<off>	<off>	<off>	<off>	<off>	<off>	D	E1	<off>	<off>	<off>	<off>	<off>
399510055														
399577605														
399532310														
399576231	E1	D3	<off>	<off>	E1	D	D	<off>	<off>	D3	D	E1	<off>	<off>
399504383	D	D	D	E1	E1	<off>	<off>	D	E1	E	<off>	<off>	E1	E1
399522121	E1	D	<off>	<off>	<off>	<off>	<off>	<off>	<off>	E1	D	<off>	<off>	<off>
399500880	<off>	E1	D3	D8	D8	<off>	<off>	N	N	E1	<off>	<off>	E1	E1
399544861	<off>	<off>	<off>	E1	D	D	D	<off>	<off>	E1	N	N	<off>	<off>
399530839														
399590468	D	E	E1	D	<off>	<off>	<off>	D8	E1	E	D3	D3	<off>	<off>

Results

Software

- Software for solving this class of NRPs was built on top of SINTEFs SCOOP library that contains a CP/CSP library.
- The development was an industry project financed by a software vendor:
  - Should "always" work - needed robust method over large variety of problems
- Development started as early as 2004, first deployment in 2006.
- Research and software development have continued since.
- The system is currently in use in several Norwegian hospitals.
- Interest outside hospitals (newspapers, counties) + interest in Sweden.
- Currently we're developing a more generic model, capable of handling generic pattern and work load constraints.

Results

- Huge problems solved; up to 80 employees and 168 days. Typically 9 different shift types, in three categories.
- It is necessary with a powerful diversification step when doing (iterated) local search – CP is a good tool.
- The use of CP quickly finds a feasible solution in the first phase, and in the rebuild phase.
- CP is efficient;
  - Initial solution found in a few seconds on small to medium cases, up to one minute on the largest cases (~2 mill backtracks)
  - In the rebuild of the diversification step usually a fraction of a second is used.
- We can solve a large range of real-world NRPs in reasonable time.

Thank you for your attention!

- Detailed model description (MIP style) can be found at: http://www.comihc.org/index.php/Models/sintef-ict-nurserostering-model.html

- The test cases can be found at: http://www.comihc.org/index.php/Test-Beds/sintef-ictnurse-rostering-data.html