Solving a nurse rostering problem with enhanced tabu search

Dr Kath Dowsland

Outline

- The problem
- Choice of approach
- Potential difficulties
- Building a solution
- Final Outcome.
The problem.

To produce weekly schedules of work for all nurses on the ward so that:

- minimum covering requirements are met
- nurses’ preferences and requests are considered
- schedules are deemed to be fair

The day:
A day is made up of 3 shifts:
- earlies (7.5 hour day)
- lates (7.5 hour day)
- nights (9.5 or 9 hour night)

Nurses work either a whole week of days (usually a mixture of earlies and lates) or a whole week of nights.

Types:
A full-time nurse works 5 days or 4 nights.
Part-timers work other combinations
  e.g. (4,3), (3,3), (3,2).
Days off and study days yield yet more combinations.
Grades:
There are 3 grade bands.
Covering requirements are given cumulatively for each band.

Patterns and costs.
Each nurse type defines a set of feasible shift patterns.
A full time nurse defines 21 feasible day patterns and 35 night patterns.
Each nurse-pattern pair has a penalty cost given by:
• general quality of pattern
• meeting of requests for days off
• history of patterns worked recently

The problem is tackled in 3 stages.

1. Determining whether or not there are sufficient nurses and adding bank nurses if not.

2. Finding a feasible solution which minimises preference costs.

3. Allocating day shifts to earlies and lates.
Stage 2.

Notation.

\( x_{ij} = 1 \) if nurse \( i \) works pattern \( j \), = 0 otherwise.

\( p_{ij} \) is the penalty associated with nurse \( i \) working pattern \( j \)

\( F(i) \) is the set of patterns feasible for nurse \( i \).

\( a_{ik} = 1 \) if pattern \( j \) covers shift \( k \)

\( G_r \) is the set of nurses of grade-band \( r \) or above

\( R(k,r) \) is the minimum acceptable number of nurses of grade \( r \) or above for shift \( k \).

The problem can then be written:

\[
\begin{align*}
\min z &= \sum_{i=1}^{8} \sum_{j \in F(i)} p_{ij} x_{ij} \quad (1) \\
\text{s.t.} \quad &\sum_{j \in F(i)} x_{ij} = 1 \quad \forall i \quad (2) \\
&\sum_{i \in G_r} \sum_{j \in F(i)} a_{ik} x_{ij} \geq R(k,r) \quad \forall r, k \quad (3) \\
&x_{ij} = 0 \text{ or } 1 
\end{align*}
\]

Exact solution not considered because:

- the I.P formulation will have around 1400 binary variables and 42 constraints,
- the penalties are based on subjective decisions
- further subjective judgement may be necessary to distinguish between two good solutions.
- the solution approach must be flexible in case of changes to problem specification
Local search is a good candidate because

- it has a proven track record on other multi-objective scheduling problems
- it will produce different solutions from different random number streams
- it does not rely on a linear objective / constraints

Potential problem.

Local search heuristics such as simulated annealing and tabu search are essentially blind and work best on a smooth undulating landscape. They are not good at:

- Climbing over large ‘hills’
- Moving over plateaus
- Moving through sparsely connected regions
- Exploring the boundaries of the feasible region

These features often occur in real-life problems such as ours.
The basic tabu search ingredients of:

- A local search framework
- A tabu list
- An aspiration criterion
- A frequency based diversification strategy

Can be enhanced by:

- Modified evaluation (cost) functions
- Candidate list strategies (*restrict moves to a list of candidates having some predefined property*)
- Compound moves
- Strategic oscillation (*oscillate search between different features e.g. searching for feasible solutions and low cost solutions.*)
- Exploiting influential attributes (*identify important attributes and tailor basic / enhanced features to them.*)

Escaping over the mountains with problem specific / intelligent diversification.
Navigating flat-bottomed valleys using chains of moves.

Moving inside sparsely connected regions with compound moves.
Local search framework.

Finding a feasible solution can be difficult. Therefore relax covering constraint.

**Feasible solutions**: set of allocations of nurses to feasible shift patterns.

**Neighbourhood moves**: changing the shift pattern of a single nurse.

**Cost**: combination of covering cost and penalty cost.

Two part cost function can be tackled using weights.

**BUT**

Not very successful as we really need to be able to move close to the boundaries of the feasible region.
Phase 1. Find a feasible solution by reducing covering cost to zero.

Phase 2. Improve solution in terms of penalty cost without leaving feasible region.

Phase 3. Reduce penalty cost by crossing feasibility boundary.

Simple descent strategy.
Random descent using covering cost as evaluation function and restricting candidate list to moves which do not increase penalty cost.

Result.
- Rarely converges to zero cost solution.
- Local optima of two types.

Type 1.
Balance of day and night shifts incorrect.

Solution:
This is easily corrected by restricting candidate list to moves which move from over-covered to under-covered types. The evaluation function is the covering cost and a best selection strategy is employed.
Type 2.
Balanced local optima are frequently surrounded by plateaux and small uphill steps. Standard tabu search therefore frequently wanders aimlessly around ‘flat-bottomed valleys’

Solution:
Guide search by introducing chains of moves whose net result is a fall in covering cost and no increase in penalty cost.
Two types of chain have been implemented.

Shift-chains.
Define a chain of shifts from an over-covered to an under-covered shift. Execute the chain by changing the pattern of one nurse by one shift for each link in the chain.

| Nurse A | 1 1 1 1 1 0 0 0 0 0 0 0 0 |
| Nurse B | 0 1 0 0 1 1 0 0 0 0 0 0 0 |
| Nurse C | 1 1 0 1 1 0 1 0 0 0 0 0 0 |

Path: 3 → 7 → 1 → 6
Problem can be modelled as that of finding zero or negative cost paths in a small graph.
Edge from i to j if any nurse can move from i to j.
Cost = change in cost for cheapest such move.

Nurse-chains.
Define a chain of nurses from a nurse working an over-covered shift to a nurse who can work a shift pattern which moves the over-covering to an under-covered shift.

Path A → D → C

Problem can be modelled as that of finding zero or negative cost paths in a small graph.
Phase 1 moves.

<table>
<thead>
<tr>
<th>Neighbourhood</th>
<th>candidate list</th>
<th>evaluation function</th>
<th>selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Standard</td>
<td>no PC</td>
<td>CC</td>
<td>first decrease</td>
</tr>
<tr>
<td>2 Standard</td>
<td>balance</td>
<td>CC</td>
<td>best</td>
</tr>
<tr>
<td>3 shift-chain</td>
<td>all</td>
<td>PC</td>
<td>first non-increasing</td>
</tr>
<tr>
<td>4 nurse-chain</td>
<td>all</td>
<td>PC</td>
<td>first non-increasing</td>
</tr>
<tr>
<td>5 standard</td>
<td>over to under</td>
<td>CC</td>
<td>best</td>
</tr>
<tr>
<td>6 standard</td>
<td>all</td>
<td>-</td>
<td>first</td>
</tr>
</tbody>
</table>

Phase 2.
Reduce covering cost while remaining feasible.
Problem: standard neighbourhood leaves solution space disconnected / sparsely connected.
Random descent over 3 neighbourhoods.

<table>
<thead>
<tr>
<th>neighbourhood</th>
<th>candidate list</th>
<th>evaluation function</th>
<th>selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 standard</td>
<td>no CC</td>
<td>PC</td>
<td>first decrease</td>
</tr>
<tr>
<td>2 shift-chain</td>
<td>all</td>
<td>PC</td>
<td>first decrease</td>
</tr>
<tr>
<td>3 nurse-chain</td>
<td>all</td>
<td>PC</td>
<td>first decrease</td>
</tr>
</tbody>
</table>

When a local optima is reached a single phase 3 move reduces the penalty cost at the expense of covering cost.
Tabu lists and influential attributes.
The following additional features complete the search definition.

- A tabu list of length 1 based on the nurse last moved.

There are still problems with day/night ‘mountains’ which were solved as follows

- A tabu list of length 5 based on the day-night partitions.
- A diversification from the current day-night partition after 50 moves or sooner if partition is bounded.
- An additional swap neighbourhood to produce balanced day / night cover under certain conditions.

Outcome:
In tests on over 50 data sets the heuristic was able to match the optimal solution consistently.
The hospital were pleased with the solutions.
Since starting the project there have been several changes in problem specification including:

- Spreading over-covering
- Incorporating mixed day/night patterns
- Restricting w/e working for top grades
- Allocating nurses according to teams

All have been successfully added into the basic framework described above.