Nurse Rostering

A Practical Case

Greet Vanden Berghe
KaHo Sint-Lieven, Gent, Belgium
greetvb@kahosl.be

Introduction

- Nurse scheduling, employee timetabling, personnel rostering
- Problem in Belgian hospitals

Approach

- HARD CONSTRAINTS
  - coverage (personnel demands per day per shift type and per skill category; minimum/preferred)
- SOFT CONSTRAINTS
  - time related constraints on the personal schedules

Goal: schedule the resources to meet the hard constraints while aiming at a high quality result with respect to the soft constraints.

Characteristics

- highly constrained
- often over constrained
- not suitable for OR methods
- 'cultural' differences between countries
- the requirements of the hospitals in Belgium can not be met with a cyclical 'three shift' schedule
- the personnel prefer a schedule in which they can express their personal wishes and priorities
- the problem is mainly tackled manually in practice
- heuristics: solutions in an acceptable calculation time
  - Variable neighbourhood search
  - Tabu search
  - Genetic & Memetic algorithms
- the algorithms presented in this lecture are developed for commercial nurse rostering software for Belgian hospitals: Plane

Requirements

- Functional requirements
  - Conceptual consistency and continuity
  - Graceful degradation
  - Pertinent behaviour
  - Explanatory power
  - Extendibility
- Problem specific requirements
  - Hard constraints
    - Personnel requirements (minimum and preferred coverage)
    - Skill category (non-hierarchical replacement organisation)
  - Soft Constraints

Soft Constraints

- Hospital constraints
  - Minimum time between two assignments
- Constraints defined by the work regulation
  - Maximum number of assignments during planning period
  - Minimum/maximum number of consecutive days
  - Minimum/maximum number of consecutive free days
  - Maximum number of assignments per day of the week
  - Maximum number of assignments for each shift type
  - Maximum number of a shift type per week
  - Number of consecutive shift types
  - Assign 2 free days after night shifts

Soft Constraints

- Assign complete weekends
- Assign identical shift types during the weekend
- Maximum number of consecutive working weekends
- Restriction on the succession of shift types
- Patterns enabling specific cyclic constraints
- Balancing the workload among personnel

Personal constraints
- Day off / shifts off
- Requested assignments
- Tutorship (people not allowed to work alone)
- People not allowed to work together

Fitness Evaluation Method

- Modular
  - Implementation
  - Evaluation
- Explanatory
  - Violated constraints
  - Violation extent
- Easily extendible
  - New constraints
  - Extra features
- Modifiable
  - Cost parameters
  - Constraint values
  - Considering previous planning period

- Low memory use
  - Simultaneous calculations of different hospital wards
- Fast
  - Evolutionary algorithms
- Complex real world problems
- Simple evaluation algorithm

General variables

- planning period (usually 4 weeks) = 28 days
- shift types, time units

<table>
<thead>
<tr>
<th>Shift Type</th>
<th>Start (Half Time)</th>
<th>End</th>
</tr>
</thead>
<tbody>
<tr>
<td>Morning</td>
<td>06:45</td>
<td>11:00</td>
</tr>
<tr>
<td>Morning</td>
<td>06:45</td>
<td>14:45</td>
</tr>
<tr>
<td>Day</td>
<td>08:10</td>
<td>16:00</td>
</tr>
<tr>
<td>Evening</td>
<td>14:15</td>
<td>22:15</td>
</tr>
<tr>
<td>Night</td>
<td>22:00</td>
<td>07:00</td>
</tr>
</tbody>
</table>

- scheduled event, day off
- personnel members, skills

Real World Issues

average
- 20 personnel members per ward
- 6 shift types
- 30 types of active constraints per person
- 100 evaluations of the cost function per iteration
- 300 iterations per minute on IBM RS/6000

The evaluation function is also applicable to other timetabling and scheduling problems


Consistency check

- Difficult to define the problem in a feasible way
- Assists personnel managers or head nurses to define feasible problems
- The procedure takes hard constraints and a selection of precedence soft constraints:
  - Personal requests (days/shifts off e.g. absence, illness)
  - Personal requests (shifts)
  - Patterns
- Detected infeasibilities:
  - Too few skilled people available at a certain time
  - Too many skilled people scheduled at a certain time
- Two strategies:
  - Accept the fact that constraints will be violated
  - Repair: make the personnel requirements feasible
Tools for varying needs

- Freezing parts of the planning period (people – time)
- Quick re-scheduling in emergency situations
- Scheduling some personnel members manually
- Formulating the personnel requirements in terms of time intervals (floating) instead of shift types
  - Translate the time interval requirements in shift type requirements
  - Allow changes of shift type combinations during the search

Initialisation

- initial solution: feasible schedule, satisfying the hard constraints
  - three strategies to start:
    - Current schedule (time when the requirements have changed after the schedule was made)
    - Previous planning period (time when the requirements and constraints are similar)
    - Empty schedule (often most efficient choice)
  - Make the schedule feasible by randomly adding or removing duties for every category until the personnel requirements are met

Pre- and post-planning options

PRE-PLANNING
- Set the hard constraints equal to:
  - Preferred requirements
  - Minimum requirements

POST-PLANNING
- Add shift types towards the preferred requirements (only when minimum requirements were selected for the planning)
- Add shift types to reduce undertime (this option allows violations of hard constraints)
  - Never add shift types for which the minimum requirements are 0
  - Never add night shifts

Planning options

- Quick planning
  - Applies basic algorithms for quickly generating an acceptable schedule
- Thorough planning
  - Applies hybrid algorithms with human-inspired improvement techniques
- Extra thorough planning
  - With additional algorithms to solve replacement between skill categories

Solution Approaches

- Algorithms are not made-to-measure
  - Minimising the value of the cost function without awareness of the components of that cost function
- Steepest descent
- Meta-heuristics
  - Hybrid tabu search (with different neighbourhoods)
    - Dynamically change the search heuristics
    - Address particular constraints on the problem
    - Keep the calculation time down
    - Some of the neighbourhoods of previous approaches have been re-used
  - Memetic, genetic, simulated annealing, …

Some Meta-heuristic approaches

- Hybrid tabu search algorithm
  - Implemented in the commercial software

- Memetic algorithms
  - Interesting results but too time consuming for implementation in practice
Hybrid Tabu Search algorithm

- Tabu search algorithm
  - No violation of hard constraints during the search
  - Stop criterion

- Hybridisations
  - Applied in different neighbourhoods
  - Solving particular soft constraints

Single shift-day neighbourhood

- Simplest neighbourhood
- Position of one assignment differs
- Demonstration of a move

<table>
<thead>
<tr>
<th></th>
<th>Mon</th>
<th>Tue</th>
<th>Wed</th>
<th>Thu</th>
</tr>
</thead>
<tbody>
<tr>
<td>Head Nurse</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
</tr>
<tr>
<td>Nurse A</td>
<td>E</td>
<td>N</td>
<td>E</td>
<td>L</td>
</tr>
<tr>
<td>Nurse B</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Nurse C</td>
<td>E</td>
<td>N</td>
<td>E</td>
<td>E</td>
</tr>
</tbody>
</table>

Original tabu search algorithm

- A move takes an assigned shift type from one person to another on the same day
- The move is not allowed if
  - The person does not belong to the skill category of the duty
  - There is already an assignment for that shift type
- The hard constraints remain fulfilled by this kind of move
- Iterative calculations in each new environment:
  - Calculate the best possible move which is not forbidden by the tabu list (ignore the tabu list when the move leads to a better solution)
  - Perform this move
  - Add characteristics of the move to the tabu list
- The results are better than those of the steepest descent algorithm but they are not satisfactory in practice

Some heuristics for the problem

- Soft constraint related neighbourhood: complete weekends
  - The cost parameter of the soft constraint on complete weekends can be modified by the users but this constraint often causes problems
  - Solve the constraint the hard way, without taking care of the other soft constraints
- Shuffle: solve the worst personal schedule
  - Increase the quality of the worst personal schedule by exchanging a part of this schedule with a part of another person’s schedule
- Greedy shuffling: Model human-inspired scheduling techniques
  - Calculate all shuffles for all the personnel in the schedule
  - List these moves (highest benefit of the cost function first)
  - Perform as many moves as possible
Soft constraint related neighbourhoods

- weekend neighbourhood
- overtime-undertime
- alternative skills
- personal requests
- most violated constraint
- etc...

Swapping large sections of personal schedules (1)

- weekend shuffling

Swapping large sections of personal schedules (2)

- greedy shuffling
  - shuffling between all pairs of personal schedules
  - shuffling length from 1 day till half the planning period
  - Inspired by human improvements

Swapping large sections of personal schedules (3)

- core shuffling
  - greedy shuffling within greedy shuffling

Tabu Search algorithm in the commercial software

- Quick planning
  - move
  - weekend step/shuffle
  - repeat until stop criterion is met

- Thorough planning
  - Quick planning
  - finally: greedy shuffle

- Extra thorough planning
  - thorough planning
  - restart with different planning order for the skill categories

reliable solutions
short calculation time
quick check of the constraints

high quality solutions
the users evaluate the solutions often better than the cost function indicates

Excellent solution for bad skill category order
Long calculation time
### Test results

Hybrid Tabu Search

<table>
<thead>
<tr>
<th>Problem</th>
<th>Min Cost</th>
<th>Min Time</th>
<th>R-Calc Cost</th>
<th>R-Calc Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Steepest descent</td>
<td>1338</td>
<td>44</td>
<td>1144</td>
<td>47</td>
</tr>
<tr>
<td>Tabu search</td>
<td>1189</td>
<td>57</td>
<td>933</td>
<td>1135</td>
</tr>
<tr>
<td>Quick</td>
<td>843</td>
<td>18</td>
<td>86</td>
<td>24</td>
</tr>
<tr>
<td>Hybrid</td>
<td>809</td>
<td>25</td>
<td>581</td>
<td>19</td>
</tr>
</tbody>
</table>

### Genetic and Memetic Algorithms

- **GA (crossover - mutation)**
  - Problem to find a genetic presentation of the problem so that ‘characteristics’ can be inherited
- **memetic**: steepest descent on each individual, random planning order of the qualifications
- **hybrid memetic**: tabu search (or hybrid tabu search) on each individual

### Crossover - Mutation

- best personal schedule from each of the parents; random selection of the other personal schedules from the parents
- random selection of personal schedules from the parents
- select per person the best schedule from the parents
- select the best x events in the schedule
- select the best x events per person in the schedule
- first part of the planning period (before a day randomly chosen per person) from one parent, second part from the other parent
- ...

### Some examples of crossover operators for the nurse rostering problem

#### Original Memetic Algorithm: M

- Steepest descent for each individual
- Applies the ‘move’ discussed earlier
- Offspring: 2 new individuals per parent pair
- Genes: Entire personal schedules
- Newly generated schedules are not feasible ⇒ change the coverage to make them feasible

#### Original memetic algorithm: M

- Create N different schedules using random initialization while stop criterion is not reached (stop criterion: no improvement during two generations), repeat
  - Make all the schedules feasible by randomly adding and deleting appropriate shifts
  - Perform crossover:
    - Select parents from the individuals by tournament selection
    - Select parents per pair of parents, genetic children:
      - child 1:
        - Best personal schedule (tie from parent 1)
        - Best personal schedule (tie from parent 2)
      - For the other child, to be tournament selection from parent maps
      - child 2:
        - Best personal schedule (tie from parent 8)
        - Best personal schedule (tie from parent 9)
        - (check to see if the same tie is best for both parents)
        - For the other child, to be tournament selection from parent maps

---

*Diagram of the genetic and memetic algorithms*

*Some examples of crossover operators for the nurse rostering problem*

*Original memetic algorithm: M*
DM: diverse memetic algorithm
- The same ideas as in M
- The planning order of the skill categories is randomly chosen

DMR: diverse memetic algorithm with random selection
- The same as DM
- random selection of personal schedules

MSR: Memetic algorithm with string recombination
- Based on DM
- Offspring takes personal schedules partially from both parents
- Crossover per personal schedule at a random day/shift

MEx: Memetic algorithm copying the x best events
- Based on DM
- Copying x best assignments from each parent

Combining hybrid tabu search & evolutionary approaches

TSPOP: different initial solutions and randomising the planning order of skill categories
- quick hybrid tabu search on each individual
- greedy shuffling on the final solution

MEH: memetic algorithm with human-inspired improvement functions
1. create N random individual solutions while the stop criteria is not reached
   (stop criterion: no improvement during two generations), repeat steps 2 - 5.
2. make all the schedules feasible by adding and removing appropriate shifts randomly
3. perform the TSI for each of the sub-schedules
   (choose the planning order of the skill categories at random)
4. select parents from the individuals by tournament
5. recombine the parents as explained in the NGA algorithm
6. perform the greedy shuffling step on the best individual

SWT: switch
- The only algorithm in which the coverage can change
- Based on ME4 (MEx with x=4)
- Random mutation by adding or removing assignments within the feasible domain (between minimum and preferred coverage)

Comparison of test results
Conclusion

- Automating the nurse rostering problem:
  - Reduction of scheduling effort and time
  - Quick evaluation of schedules, taking care of all the constraints
  - Manually adaptable
  - Good quality of solutions; objective schedules

- Combining tabu search with some specific problem-solving heuristics: Hybrid Tabu Search
  - Better quality of solutions
  - Acceptable increase in calculation time

- Genetic:
  - Calculation time about 100 times longer than hybrid TS
  - The same quality of solutions

- Memetic:
  - Calculation time: 10 times longer than genetic
  - Results: about 10% better
  - Less dependent on initialization and parameter changes