Multi-skill shift design for Norwegian hospitals

Eivind J. Nilssen, Martin Stelevik, Erik Lien Johnsen and Tomas Eric Nordlander *

SINTEF ICT, Department of Applied Mathematics, Oslo, Norway

Abstract. The shift design problem consists in finding a suitable set of shifts which covers the work demand for a planning horizon. For labor intensive organizations, finding a good match between the predicted workload and the scheduled workforce capacity is crucial. In a multi-skill shift design problem, the model must reflect the skill attributes of the employees and the time-dependent demand of each skill type. For hospitals, a number of constraints and objectives complicate the picture. In this paper we introduce models which reflect many of the challenges faced by planners at two Norwegian reference hospitals. Experiments using mixed-integer programming solvers show promising results, and near-optimal solutions are found within a few minutes.

Keywords: multi-skill shift design; mixed-integer programming; workforce management

Introduction

In developed countries, population ageing along with the skewed demographic development, result in a rapid increase of hospital patients: This combined with the public’s rising expectations of quality treatment and the desire to reduce the health expenditure, all add to the strain on the hospital resources (personnel, equipment, etc.). Today the planning and scheduling of resources is to a large degree handled manually—this leads to inefficient utilization and unnecessary admin work for the medical practitioners.

Workforce management (WFM) involves matching workload with workforce, which is vital for the labor intensive health sector. A mismatch leads either to longer waiting times or increased labor cost. The WFM process is often divided into: workload prediction; staffing; shift design and/or scheduling; and staff rostering. Note that by staff rostering, we mean the short term timetabling of staff, not to be confused with other uses of the term ‘rostering’ across the literature (Burke et al. 2004). Hospital planners create staff rosters by assigning hospital employees to given shifts while considering skills, fairness, laws, regulations, etc. This combinatorial explosive problem is often handled manually by highly qualified health personnel—time-consuming and with inefficient rosters as a result.

A prerequisite to high quality nurse rosters is an efficient shift design, which aims to find a set of shifts to match predicted work demand for a planning horizon while satisfying a set of constraints. Good shift design is important for the hospital (e.g. lowering cost, managing staff coverage, staff satisfaction) and its staff (e.g. healthy working shifts). The number of possible solutions for the shift design problem increases exponentially with the size of the problem and quickly becomes too large for an efficient manual approach. Optimization algorithms excel on these
types of combinatorial explosive problems: Searching for optimality by comparing feasible solutions against each other upon certain criteria (e.g. minimization of personnel cost) to highlight the optimal one(s).

This paper presents a prototype for automatic multi-skill shift design, where shift details (start and end times, required staffing for each shift and each skill type) form the solution to an optimization problem. Our idea of automating shift design originates from workshop discussions with AHUS (Akershus universitetssykehus HF) and SAB (Sykehuset Asker og Bærum HF), two Norwegian hospitals in the Oslo region. They explained that manually creating the shifts was not only time-consuming work but they were unsure about their quality. They meant further that specifying the required input for the shift design problem was relatively straightforward. These two hospitals have supported the development process throughout with ideas, criticisms, and data provision.

We have used mixed-integer programming (MIP) as our solution method and Xpress-MP as the solver, which is linked with Microsoft Excel for rapid prototyping. Excel is also our GUI (holding data, parameter values, and visualizing solutions), with the planning personnel at both hospitals being experienced Excel users. The mathematical models were constructed through an iterative development process with our reference hospitals. They include several user specific constraints, such as: a maximum of different shifts to be used in given periods; overlap between shifts; a common leaving time for all day shifts; one main shift having most of the workers assigned to it. Our model allows for work tasks with skill requirements, which must be properly matched by skilled staff. The models’ main objectives are: minimization of personnel cost and number of different shifts in use. The main contributions of this paper are:

1. We target the important research domain of Shift Design—a prerequisite for finding high quality nurse rosters.
2. We introduce a prototype for automated multi-skill shift design for hospitals that reduces labor costs.
3. Our model is developed in close cooperation with Norwegian hospitals.
4. Our test cases are synthetic instances but they derive from detailed discussions with the hospitals. With this article, we make the test cases public to allow for reproduction of our experiments (Nilssen et al. 2010a).

Note that this paper is an invited extended version of an existing paper (Nilssen et al. 2010) presented at 2nd International Conference on Applied Operational Research in Turku, Finland. The experiments presented in the previous paper and in this show very good results. Within ten minutes we find a solution with an objective value gap to the best lower bound (obtained from LP relaxations) usually below 1 % for multi-skill problems. For single-skill problems optimal solutions are found within seconds or a few minutes.

The paper is organized as follows: Section 2 provides the background information and related research. The problem is introduced and formally stated in section 3. In section 4, we describe the experiment setup and discuss the results. We conclude and present the planned future work in section 5.

Background

For labor intensive organizations, finding a good match between the predicted workload and the scheduled workforce work capacity is crucial. Mismatch leads either to delays or unnecessarily expensive labor costs. This matching process is part of the workforce management (WFM) process and is often viewed as four phases (Musliu et al. 2004; Bhulai et al. 2008):

- **Predicting workload**: Finding the demand for personnel.
- **Staffing**: Making long term management decision of how many people to employ.
- **Shift design** (also called **shift scheduling**): Generating shifts that include start and end times as well as the number of (skilled) people required for each shift.
- **Personnel scheduling/rostering**: Assigning employees to given shifts and days off.

These steps are all interrelated, and should ideally be treated simultaneously, but due to tractability they are often treated sequentially. For the hospitals, these steps would read as; listing the staff tasks (prediction of workload); estimating their required working time (staffing); designing shifts and scheduling them to minimize the mismatch (shift scheduling); and lastly assigning employees to the shifts ( rostering).

In our reference hospitals, most time is spent on the personnel scheduling part. The research community have had the same focus, providing a large body of literature on nurse rostering (Burke et al. 2004) and personnel scheduling (Ernst et al. 2004) over the last forty years. Less attention has been given to the shift design phase,
however over the last decade the interest in this topic has increased. Note that efficient shift design is a prerequisite for high quality nurse rosters.

**Shift design**

Shift design arise in many labor intensive organizations, such as airports, call centers, transportation, health care, etc. The process of shift design consists of generating a set of shifts (which fulfills requirements on minimum and maximum shift duration, legal start and end times, etc.), with an associated staffing demand assigned to each shift, so that the time-dependent and skill-dependent demands are satisfied at all time periods. The personnel demands supplied as input to the shift design problem vary between days of the week, typically covering 24h with 15 minutes intervals. Finding an optimal shift design involves comparing different solutions given user specific objectives, such as minimizing cost and minimizing the number of shifts in use. Typical constraints relate to the matching of staffing demands and time-related shift attributes (duration, allowed start and end times, etc.). For hospitals and other organizations where transfer of information is crucial, additional constraints such as overlap between shifts may be required.

For real size problems the solution space is too large for an efficient manual approach: Even though manually finding a solution is possible, it is unlikely to be the optimal solution. Still, shifts are in most Norwegian hospitals designed manually.

**Related work**

The literature on shift design is sparse (compared to staff scheduling) and driven by the growth of call-centers, in which the personnel cost is the predominant cost at 60-70% (Fukunaga et al. 2002). For hospitals, a range of solution methods have been investigated on the shift design problem: For example, constraint programming and integer programming (Bhulai et al. 2008); mixed-integer programming and tabu search (Musliu et al. 2004; Canon 2005); and hybrid algorithms (Gaspero et al. 2010).

Our research have similarities to (Musliu et al. 2004), but differs on several points. Firstly, while they focus on solving the general shift design problem, we focus on modeling and solving the shift design problem for hospitals. Secondly, their attention is on the algorithm and how it works, while our focus is on modeling the problem and developing a prototype for hospitals. Thirdly, their work is on single skilled shift design while ours is on multi-skill shift design. Lastly, they use local search, an approximate solution method that cannot guarantee finding optimality (but is known to often find good solutions in a relatively short time). We are using MIP, an exact solution method, and we manage to find optimality within seconds for several of the cases.

Another closely related work is (Bhulai et al. 2008). As we do, they also look at multi-skilled shift design and we both apply exact solution methods. Their paper centers on presenting a new method while ours also presents a prototype. Another difference is that they focus on solutions for call centers, while we are looking at the hospital domain. Lastly, theirs is a generalized model while ours is tailored with more constraints.

In (Gaspero et al. 2010) a hybrid solution method was used to solve the combined shift design and break problem—Local Search method for determining the shifts and Constraint Programming for assigning breaks. We tackle only shift design and use a complete solution method while they use an incomplete solution method. Both models try to find a solution that minimize cost for deviation from the workload requirement and use a weight to reduce number of shifts. However, our model incorporates other objectives as well (e.g. joint day-shift end time; match the amount of man-labor year, etc.). In addition we incorporate a set of—for our reference hospitals—required constraints.

A lot of the research on shift design involves scheduling of breaks during the shifts. We refer to the central work of (Aykin 1996) in which the problem is modeled as a set-covering problem and integer programming is used to solve it. Optimal break placement is also central in (Gaspero et al. 2010). In Norwegian hospitals breaks are not scheduled and hence not included in our model.
Mathematical model

Our prototype designs shifts (start and end times) and assigns a number of workers to these shifts, thereby satisfying the demand of workers for each time slot. The demand is given as the number of employees needed, for each skill type and each time period (every 15th minute during the entire day), for all weekdays. Furthermore, the workforce can be subdivided into groups based on skill attributes. All workers in a skill group have exactly the same set of skills. The cost of having an employee assigned to a shift is given for each group and each time slot. The cost is different for time slots belonging to daytime, evening, and night intervals.

We now present the mixed-integer programming models used to design the shifts. We start with the simplest model that minimizes the cost while satisfying the demand. Later we present the modifications needed in order to take into account additional realistic constraints and objectives.

The indices and sets are:

- \( s \)  
  shift (defined by a start and end time).
- \( d \)  
  day.
- \( c \)  
  skill.
- \( g \)  
  group of workers, all with the same set of skills.
- \( S \)  
  set of possible shifts.
- \( S_t \)  
  set of shifts that cover time slot \( t \).
- \( D \)  
  set of days, \{Monday, Tuesday, ..., Sunday\}.
- \( C \)  
  set of skills.
- \( G \)  
  set of groups.
- \( G_C \)  
  set of groups that have all of the skills in \( C' \subseteq C \). In other words: workers belonging to any of the groups in \( G_C \) have at least all of the skills in \( C' \).

The parameters are:

- \( K_{sdg} \)  
  the cost of one member of group \( g \) working shift \( s \) on day \( d \).
- \( B_{tdc} \)  
  the demand for skill \( c \) in period \( t \) on day \( d \).

The decision variables are:

\( x_{sdg} \in \{0, 1, \ldots\} = \) the number of members from group \( g \) working shift \( s \) on day \( d \).

The model, in its simplest form, is then:

\[
\text{minimize} \quad \sum_{s \in S, d \in D, g \in G} K_{sdg} x_{sdg} \\
\text{such that} \quad \sum_{s \in S_t, g \in G_c} x_{sdg} \geq \sum_{c \in C'} B_{tdc}, \quad t \in T, d \in D, C' \subseteq C
\]

Constraint 2 says that the demand must (at least) be met in all time periods, on all days and for all sub sets of skills. This generates a lot of constraints, out of which many are redundant and are removed before entering the solver stage. Before we examine the more extended model we need to introduce additional indices and sets:

- \( i \)  
  an interval of the day (day, evening or night).
- \( s' \)  
  shift.
- \( I \)  
  the set of intervals for 24 hours: \{day, evening, night\}.
- \( S_{\text{day}} \subset S \)  
  the set of day shifts.
- \( S_M \)  
  the set of shifts that may become the Main shift, which is a subset of \( S_{\text{day}} \).
- \( S_{\text{evening}} \subset S \)  
  the set of evening shifts.
\[ S_{\text{night}} \subseteq S \]

the set of night shifts.

\[ T_E \]

the set of possible end times for day shifts.

\[ S_{t}^E \]

the set of shifts ending at the end of time period \( t \).

**Additional parameters are:**

- \( z_{ig} \): the maximum allowed number of shifts used in one day in interval \( i \) for group \( g \).
- \( \eta \): the fraction of the day workers which should be on each main shift.
- \( n \): the maximum number of different main shifts allowed per day. \( n \) is typically set to 1.
- \( \tau_s \): the length of shift \( s \).
- \( M_1^{sg}, M_2^{sdg}, M_3^{s'dg}, M_4^{td} \) are different big \( M \) constants.

**Additional help variables are:**

- \( y_{sg} \in \{0, 1\} = 1 \) if shift \( s \) is used by group \( g \); 0 otherwise.
- \( z_{sdg} \in \{0, 1\} = 1 \) if shift \( s \) is used by group \( g \) on day \( d \); 0 otherwise.
- \( \epsilon_{td} = 1 \) if some shifts are ending at the end of time period \( t \) on day \( d \); 0 otherwise.
- \( n_E \) is the maximum number of shifts that can end at the same time on a day.

We may have objective components other than minimizing the total cost. The following objective is minimizing the number of shifts used:

\[
\text{minimize } \sum_{s \in S, g \in G} y_{sg}
\]  

(3)

We also have to add the following constraints to model \( y_{sg} \) and \( z_{sdg} \):

\[
\sum_{d \in D} x_{sdg} \leq M_1^{sg} y_{sg}, \quad s \in S, g \in G
\]  

(4)

\[
x_{sdg} \leq M_2^{sdg} z_{sdg}, \quad s \in S, d \in D, g \in G
\]  

(5)

The effect of this ‘big \( M \)’ notation is that \( y_{sg} \) and \( z_{sdg} \) can be zero only when all \( x_{sdg} \) in the corresponding constraints are zero. The following constraints are added to limit the number of day, evening, and night shifts used for each group, each day:

\[
\sum_{s \in S_{t}} z_{sdg} \leq \tilde{z}_{i,g}, \quad i \in I, d \in D, g \in G
\]  

(6)

Constraints demanding at least \( \eta \) percent of the workers to work on the main shifts and maximum \( n \) main day shifts for each group, each day:

\[
x_{s'dg} - M_3^{s'dg} (z_{s'dg} - 1) \geq \eta \sum_{s \in S_{\text{day}}} x_{sdg}, \quad s' \in S_M, d \in D, g \in G
\]  

(7)

\[
\sum_{s' \in S_M} z_{s'dg} \leq n, \quad d \in D, g \in G
\]  

(8)

Constraints for common end time, i.e. all day shifts ending at the same time:

\[
\sum_{t \in T_E} \epsilon_{td} \leq n_E \quad \forall d \in D
\]  

(9)
\[
\sum_{s \in S_{d}^g, t \in T} x_{sdg} \leq M_{tdc}^d \quad \forall t \in T, d \in D
\]

\[
x_{sdg} = 0 \quad \forall s \in S_{day} \setminus \bigcup_{t \in T, g} S_{t}^g
\]

\[
e_{td} \in \{0,1\} \quad \forall t \in T, d \in D
\]

Constraint 11 states that day shifts not ending on possible end times \(t \in T_E\) cannot be used.

Constraint to ensure overlap between shifts:

\[
\sum_{s \in S_{d}} x_{sdg} < \sum_{s \in S_{t}} x_{sdg} \quad \forall t \in T, d \in D, g \in G
\]

Constraint 13 states that there must be more people at work in each time period than those leaving in the end of that time period. If it is not necessary to do the tasks in exactly the specified period, the demand should not be hard constraints. A way to do this is to allow for changes in demand, within some limits, by modifying the demand constraint 2. First we introduce some additional parameters and help variables.

Additional parameters:

\[
\delta_{c}^{+} \quad \text{the maximum allowed increase in demand for skill } c \text{ in any period.}
\]

\[
\delta_{c}^{-} \quad \text{the maximum allowed decrease in demand for skill } c \text{ in any period.}
\]

\[
b_{c} \quad \text{the minimum demand for skill } c \text{ after a decrease in any period.}
\]

\[
\Delta_c \quad \text{the maximum change in demand for skill } c \text{ allowed during one day. Note: In the code we only allow for demand changes for skill level 1.}
\]

Additional help variables:

\[
\delta_{tdc}^{+} \quad \text{the increase in demand for skill } c \text{ at time } t \text{ on day } d.
\]

\[
\delta_{tdc}^{-} \quad \text{the decrease in demand for skill } c \text{ at time } t \text{ on day } d.
\]

The soft demand constraints model then becomes:

\[
\sum_{s \in S_{d}, g \in C} x_{sdg} \geq \sum_{c \in C'} (B_{tdc} + \delta_{tdc}^{+} - \delta_{tdc}^{-}) \quad \forall t \in T, d \in D, C' \subseteq C
\]

\[
\delta_{c}^{+} \geq \delta_{tdc}^{+} \geq 0 \quad \forall t \in T, d \in D, c \in C
\]

\[
\min (\max(0, b_{tdc} - b_{c}), \delta_{c}^{-}) \geq \delta_{tdc}^{-} \geq 0 \quad \forall t \in T, d \in D, c \in C
\]

\[
\sum_{t \in T} \delta_{tdc}^{-} \leq \Delta_c \quad \forall d \in D, c \in C
\]

\[
\sum_{t \in T} (\delta_{tdc}^{+} - \delta_{tdc}^{-}) \geq 0 \quad \forall d \in D, c \in C
\]

Constraints 15 and 16 give limitations on the change in each period (the min function in the latter ensures that the demand is not set below a certain limit). Constraint 17 limits the total change each day, and Constraint 18 makes sure that the total demand after the changes is not lower than before.
**Empirical studies**

Our case study refers to the shift design problem faced by our reference hospital in Oslo, AHUS. We conduct several experiments concerning variations of the model. Models were implemented in MOSEL and solved by XPRESS-MP 19.00 on an Intel Xeon 3.0 GHz microprocessor. A simple GUI in Excel was developed to demonstrate the prototype and enable discussions with the end users. The experiments are based on synthetic parameter values and 24-hour demand data, with realistic values set based on input from our reference hospital (Nilssen et al. 2010a).

The present lack of suitable high-quality real 24-hour demand data has forced us to create synthetic data together with our reference hospitals. The hospitals have plans for improving and standardizing their 24-hour demand data. According to our data, the workforce (nurses, head nurse, and so on) is subdivided into 5 groups, each group having the same skill set. The skills in a skill-set can be of 5 different types. For hierarchical skill groups, the lowest group masters only one skill type, whereas the most advanced worker group masters all skill types. A slightly more difficult problem with arbitrary skill configurations among the groups is also supported by our implementation, but the experiments reported below are for hierarchical skills. The demand data we use is a 24-hour demand curve which details, for every 15th minute, the demand for each skill type. In our test cases, the 24-hour demand is the same for Monday – Thursday, slightly less for Fridays, and less still for Sat – Sun. Figures 1a – 1b display the demand curves (Monday - Thursday) for two different cases with variations in skill demand. The data is made available as benchmarks for download (Nilssen et al. 2010a).

![Fig. 1a. 24h demand curves for Monday - Thursday, for employees with four different skill levels.](image1)

![Fig. 1b. 24h demand curves for Monday - Thursday, for employees with four different skill levels, with more time variation in the skill demands when compared to Figure 1a.](image2)
Shifts designed to cover the demand must satisfy some time-related constraints, see Table 1. The table also defines the skill- and time-dependent cost of scheduling workers, plus other parameters related to the solver. All shift durations are integer multiples of 30 minutes. In a pre-processing stage, the tool generates all possible shifts adhering to these constraints.

Table 1. Parameters for the solver.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shift constraints</td>
<td></td>
</tr>
<tr>
<td>Earliest shift start</td>
<td>07:00</td>
</tr>
<tr>
<td>Latest (night) shift start</td>
<td>22:30</td>
</tr>
<tr>
<td>Earliest (night) shift end</td>
<td>07:00</td>
</tr>
<tr>
<td>Latest shift end</td>
<td>00:00</td>
</tr>
<tr>
<td>Minimum shift duration (h)</td>
<td>6</td>
</tr>
<tr>
<td>Maximum shift duration (h)</td>
<td>10</td>
</tr>
<tr>
<td>Hourly daytime cost (NOK), skills 1-5</td>
<td>187/210/220/238/245</td>
</tr>
<tr>
<td>Hourly extra evening cost, skills 1-5 (17:00-22:00)</td>
<td>50/63/68/73/78</td>
</tr>
<tr>
<td>Hourly extra night cost, skills 1-5 (22:00-07:00)</td>
<td>80/84/90/96/108</td>
</tr>
<tr>
<td>Hourly extra weekend cost, skills 1-5</td>
<td>40/52/60/65/90</td>
</tr>
<tr>
<td>Objective weights</td>
<td></td>
</tr>
<tr>
<td>Monetary cost</td>
<td>0,001</td>
</tr>
<tr>
<td>Number of shifts</td>
<td>1</td>
</tr>
<tr>
<td>Common end (as a soft constraint)</td>
<td>1</td>
</tr>
<tr>
<td>Maximum runtime (s)</td>
<td>600</td>
</tr>
</tbody>
</table>

Parameter values for the constraints in our mathematical model are detailed in Table 2. We also tested models where there is a maximum number of shifts in use within the three intervals day, evening, and night. For these cases, we set the maximum to 3 day shifts, 2 evening and 2 night shifts, per day and per skill group.

Table 2. Details regarding the problem constraints.

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overlap</td>
<td>At least one worker must overlap at least 30 minutes with the next shift. Same skill of the overlap is not required.</td>
</tr>
<tr>
<td>Common end</td>
<td>Shifts that end between 12:00 - 18:00 must end at the same time, to allow for efficient information transfer.</td>
</tr>
<tr>
<td>Main shift</td>
<td>At least 70% of daytime workers must be assigned to this shift. It must start between 07:00 - 09:00, with duration 7 - 8.5 hours.</td>
</tr>
</tbody>
</table>

Table 3.- Table 5. summarize the results of our experiments on these cases. All reported cases are for hierarchical multi-skill problems, and a basic objective of minimizing the weighted sum of monetary cost and the number of shifts in use. The tables report the plan costs, number of shifts in use, and percentage gaps to lower bounds for the different configurations of objectives and constraints. For these multi-skill problems, our solver never finds the optimal solutions within 10 minutes runtime; although the optimality gap is less than 1% for most configurations, except for the cases with a Main shift constraint. We note that for the hardest problem in a single skill setting, the solver finds the optimal solution after 630 seconds (Nilssen et al. 2010).

Table 3. Monetary cost, for seven different objectives and constraints configurations, and five data cases.

<table>
<thead>
<tr>
<th>Cost (NOK)</th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
<th>Case 4</th>
<th>Case 5</th>
<th>Average</th>
<th>St.Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic</td>
<td>327.069</td>
<td>327.037</td>
<td>326.835</td>
<td>327.142</td>
<td>324.788</td>
<td>326.574</td>
<td>1.005</td>
</tr>
<tr>
<td>Overlap</td>
<td>327.187</td>
<td>326.951</td>
<td>327.301</td>
<td>327.120</td>
<td>325.113</td>
<td>326.734</td>
<td>915</td>
</tr>
<tr>
<td>Common end (constraint)</td>
<td>347.800</td>
<td>347.744</td>
<td>347.740</td>
<td>347.744</td>
<td>346.314</td>
<td>347.494</td>
<td>662</td>
</tr>
<tr>
<td>Common end (objective)</td>
<td>326.659</td>
<td>326.741</td>
<td>326.675</td>
<td>326.531</td>
<td>324.341</td>
<td>326.189</td>
<td>1.036</td>
</tr>
<tr>
<td>Common end (obj.) + overlap</td>
<td>327.472</td>
<td>326.978</td>
<td>327.130</td>
<td>327.188</td>
<td>324.447</td>
<td>326.643</td>
<td>1.241</td>
</tr>
<tr>
<td>Main shift</td>
<td>342.564</td>
<td>340.236</td>
<td>344.874</td>
<td>340.412</td>
<td>342.877</td>
<td>342.193</td>
<td>1.923</td>
</tr>
<tr>
<td>Main + overlap + common (obj.)</td>
<td>341.389</td>
<td>359.578</td>
<td>346.247</td>
<td>343.632</td>
<td>344.560</td>
<td>347.081</td>
<td>7.203</td>
</tr>
</tbody>
</table>
Table 4. Number of shifts in the solutions, for the seven different configurations and five data cases.

<table>
<thead>
<tr>
<th>Number of shifts</th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
<th>Case 4</th>
<th>Case 5</th>
<th>Average</th>
<th>St.Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic</td>
<td>19</td>
<td>18</td>
<td>20</td>
<td>17</td>
<td>17</td>
<td>18.20</td>
<td>1.30</td>
</tr>
<tr>
<td>Overlap</td>
<td>22</td>
<td>22</td>
<td>21</td>
<td>20</td>
<td>20</td>
<td>21.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Common end (constraint)</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Common end (objective)</td>
<td>23</td>
<td>21</td>
<td>21</td>
<td>23</td>
<td>21</td>
<td>21.80</td>
<td>1.10</td>
</tr>
<tr>
<td>Common end (obj.) + overlap</td>
<td>25</td>
<td>23</td>
<td>24</td>
<td>23</td>
<td>22</td>
<td>23.40</td>
<td>1.14</td>
</tr>
<tr>
<td>Main shift</td>
<td>28</td>
<td>28</td>
<td>37</td>
<td>30</td>
<td>25</td>
<td>29.60</td>
<td>4.51</td>
</tr>
<tr>
<td>Main + overlap + common (obj.)</td>
<td>29</td>
<td>29</td>
<td>29</td>
<td>34</td>
<td>26</td>
<td>29.40</td>
<td>2.88</td>
</tr>
</tbody>
</table>

Table 5. Optimality gap, for the seven different configurations and five data cases.

<table>
<thead>
<tr>
<th>Optimality gap (%)</th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
<th>Case 4</th>
<th>Case 5</th>
<th>Average</th>
<th>St.Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic</td>
<td>0.53</td>
<td>0.41</td>
<td>0.5</td>
<td>0.37</td>
<td>0.36</td>
<td>0.43</td>
<td>0.08</td>
</tr>
<tr>
<td>Overlap</td>
<td>0.72</td>
<td>0.64</td>
<td>0.69</td>
<td>0.52</td>
<td>0.63</td>
<td>0.64</td>
<td>0.08</td>
</tr>
<tr>
<td>Common end (constraint)</td>
<td>0.43</td>
<td>0.34</td>
<td>0.45</td>
<td>0.37</td>
<td>0.55</td>
<td>0.43</td>
<td>0.08</td>
</tr>
<tr>
<td>Common end (objective)</td>
<td>0.97</td>
<td>0.84</td>
<td>0.78</td>
<td>1.1</td>
<td>0.93</td>
<td>0.92</td>
<td>0.12</td>
</tr>
<tr>
<td>Common end (obj.) + overlap</td>
<td>1.29</td>
<td>1.2</td>
<td>1.16</td>
<td>1.01</td>
<td>0.99</td>
<td>1.13</td>
<td>0.13</td>
</tr>
<tr>
<td>Main shift</td>
<td>5.9</td>
<td>5.16</td>
<td>7.01</td>
<td>5.49</td>
<td>6.39</td>
<td>5.99</td>
<td>0.73</td>
</tr>
<tr>
<td>Main + overlap + common (obj.)</td>
<td>5.84</td>
<td>10.58</td>
<td>7.05</td>
<td>6.83</td>
<td>7.38</td>
<td>7.54</td>
<td>1.80</td>
</tr>
</tbody>
</table>

Earlier experiments on minimizing cost only, resulted in much shorter runtimes where the solver quickly found optimal solutions for multi-skill problems with arbitrary skill sets (Nilssen et al. 2010). Now using a weighted sum of cost and the number of shifts in use as the objective for the solver, a considerable increase in runtime is experienced. We remark that the problem of minimizing the number of shifts in the general shift design problem is NP hard and also hard to approximate (Kortsarz et al. 2001). For tests where the Main shift constraint is allowed to vary (not shown here), the runtime increases noticeably compared to the case of a single main shift for a week. Note that depending on the demand, the Main shift constraint may force an undesirable large assignment of workers to evening shifts—due to its shorter allowed maximum duration.

Figure 2 displays the shifts generated for Monday – Thursday for case 5 (having the demand displayed in Figure 1b), using the basic configuration. Figure 3 shows the result for the same case, with the Common end constraint activated. Grey shades indicate the skill covered during the shift, where skill 1 is represented by a light grey shade and darker shades indicate skills higher up the hierarchical categories. So for instance, the worker assigned to the bottom shift in Figure 2 first performs tasks requiring skill 4, and then covers the skill 3 demand for some time, until skill 4 tasks are required before the shift ends. Cross-hatched areas indicate surplus coverage.

![Fig. 2. Designed shifts for Monday – Thursday, for case 5.](image-url)
Fig. 3. Designed shifts for case 5. The additional constraint of common leaving time for day shifts is present.

We have conducted experiments with the flexible model where demand can be moved in time, to a limited degree. Smoother demand curves should enable the solver to find better shift designs. So far, we have only experimented with modifying the demand for a single skill type. The moved demand must stay within the same day; the maximum amount of moved demand is limited to a total of 8 hours per day; and there is an upper bound on the amount of demand that can be moved in a single time slot. Early results indicate that the model works according to our intentions, with smoother demand curves and fewer shifts as a result. More input on end-user requirements is needed before further experiments on this aspect can be conducted.

Conclusion and future work

In this paper, we presented a novel approach for assisting users in both tactical as well as in strategic staff planning responsibilities. Our work focused on the development of a proof-of-concept prototype for hospital planners, with a variety of models implemented reflecting real-world constraints. We have tested our approach using instances of synthetic data based on input from our reference hospitals, and presented the results in this paper. The results show that the approach manages to solve shift design instances which are realistic both in size and in problem characteristics. The developed tool handles shift design for multi-skill demand cases given a workforce which is grouped based on employees’ skill attributes. Hospital-specific constraints are modeled, such as the design of a main shift, and a common leave time for all day shifts. Variants of these implementations are the subject of future research. We note that multi-objective optimization algorithms may be worth exploring for this problem, having multiple objective components and several soft constraints. Also, the related problem of distributing excess capacity in a robust manner, given a pre-specified workforce size, has been implemented, but more work is needed here. Along with our reference hospitals, we plan to continue our work by improving the model and developing a tool that allows them assistance in both strategic and tactical decisions.

Acknowledgments—The authors wish to thank Øyvind Bakke, Elisabeth Reitan, and Henning Moen from the hospitals AHUS (Akershus universitetssykehus HF) and SAB (Sykehuset Askør og Bærum HF) for providing us with data and their precious comments and suggestions throughout our research.
References


