Using the PEAST Algorithm to Roster Nurses in an Intensive-Care Unit in a Finnish Hospital

Nico Kyngäs[1], Kimmo Nurmi[1], Eyjólfur Ingi Ásgeirsson[2], Jari Kyngäs[1]
[1] Satakunta University of Applied Sciences
Tiedepuisto 3, 28600 Pori, Finland
[2] Reykjavík University, School of Science and Engineering, Menntavegur 1, 101 Reykjavík, Iceland

Abstract. Workforce scheduling has become increasingly important for both the public sector and private companies. Good rosters have many benefits for an organization, such as lower costs, more effective utilization of resources and fairer workloads and distribution of shifts. This paper presents a successful way to roster nurses in an intensive-care unit in a Finnish hospital. The rosters are generated using a population-based local search method called the PEAST algorithm. The algorithm has been integrated into market-leading workforce management software in Finland.

Keywords: Nurse Rostering, Staff Scheduling, Workforce Optimization, PEAST algorithm.

1 Introduction

Workforce scheduling, also called staff scheduling and labor scheduling, is a difficult and time consuming problem that every company or institution that has employees working on shifts or on irregular working days must solve. Different variations of the problem are NP-hard and NP-complete (Garey and Johnson 1979, Bartholdi 1981, Tien and Kamiyama 1982, Lau 1996), and thus extremely hard to solve. Good overviews of workforce scheduling are published by Alfares (2004), Ernst et al. (2004) and Meisels and Schaerf (2003).

Nurse rostering (Burke 2004) is by far the most studied application area in workforce scheduling. Other successful application areas include airline crews (Dowling et al. 1997), call centers (Beer et al. 2008), check-in counters (Stolletz 2010), ground crews (Lusby et al. 2010), nursing homes (Ásgeirsson 2010), postal services (Bard et al. 2003), retail stores (Chapados et al. 2011) and transport companies (Nurmi et al. 2011).

Most of the workforce scheduling cases in which academic researchers have announced that they have signed a contract with a customer concern nurse rostering (Van Wezel and Jorna 1996, Meyer auf'm Hofe 2001, Diaz et al. 2003, Kawanaka et al. 2003, Bard and Purnomo 2005, Burke et al. 2006, Bilgin et al. 2008, Beddoe et al. 2009). Hospitals tend to be very open about their operational details, enabling easy cooperation with academics who wish to publish the results of their work. However, we believe there is still a gap between academic and commercial solutions. The commercial products may not include the best academic solutions. Yet we have experienced that nurse rostering cooperation between a commercial software vendor and academics does work. According to our experience, the best action plan for real-world nurse rostering research is to cooperate both with a problem owner and a software vendor. Collaboration with software vendors and problem owners allows academics to concentrate on modeling issues and algorithmic power instead of user interfaces, financial management links, customer reports, help desks, etc.

The need for effective commercial workforce scheduling has been driven by the growth in the customer contact center industry and retail sector, in which efficient deployment of labor is of crucial importance. The balance between offering a superior service and reducing costs to generate revenues must constantly be found. There are five basic reasons for the increased interest in nurse rostering optimization. First, hospitals around the world have become more aware of the possibilities in decision support technologies and no longer want to handle the schedules manually. Second, human resources are one of the most critical and most expensive resources for hospitals. Careful planning can lead to significant improvements in productivity. Third, good schedules are very important for the welfare of the staff, resulting in increased happiness and reduction of sick-leaves. Fourth, new algorithms have been developed to tackle previously intractable nurse rostering instances, and, at the same time, computer power has increased to such a level that researchers are able to solve large-scale instances. Finally, one further significant benefit of automating the scheduling process is the considerable amount of time saved by the administrative nurses involved.

The goal of this paper is to show that the PEAST (Population, Ejection, Annealing, Shuffling, Tabu) algorithm can be used to roster nurses in Finnish hospitals. Section 2 introduces the workforce scheduling process with notes on nurse rostering. It also introduces the necessary terminology. In Section 3 we describe the characteristics of the nurse rostering problems occurring in intensive-care units in Finnish hospitals. Section 4 gives an outline of the PEAST algorithm. Section 5 presents our computational results.

2 Workforce Scheduling and Nurse Rostering

Workforce scheduling consists of assigning employees to tasks and shifts over a period of time according to a given timetable. The *planning horizon* is the time interval over which the employees have to be scheduled. Each employee has a total working time that he/she has to work during the planning horizon. Furthermore, each employee has *competences* (qualifications and skills) that enable him/her to carry out certain tasks. Days are divided into *working days* (days-on) and rest days (*days-off*). Each day is divided into periods or timeslots. A *timeslot* is the smallest unit of time and the length of a timeslot determines the granularity of the schedule. A *shift* is a contiguous set of working hours and is defined by a day and a starting period on that day along with a *shift length* (the number of occupied timeslots). Shifts are usually grouped in *shift types*, such as morning (M), day (D) and night (N) shifts. A specific sequence of shifts, such as DDDNN, is called a *stint*. Each shift is composed of a number of *tasks* that should be completed during the shift. A shift or a task requires the employee assigned to it to possess one or more competences. A work schedule for an employee over the planning horizon is called a *roster*. A roster is a combination of shifts and days-off assignments that covers a fixed period of time.

We classify the real-world workforce scheduling process as given in Figure 1. *Workload prediction*, also referred to as demand forecasting or demand modeling, is the process of determining the staffing levels – that is, how many employees are needed for each timeslot in the planning horizon. *Shift generation* is the process of determining the shift structure, tasks to be carried out on particular shifts and the competences needed on different shifts. Traditionally, hospitals work in three shifts – morning, day and night – but in the intensive-care units the customer flow should be considered when constructing the shift structure, as is the case in, e.g. the call center and retail sector businesses. The shifts generated from a solution to the shift generation problem form the input for subsequent phases in the workforce scheduling. Another important goal for shift generation is to determine the size of the workforce required to solve the demand. Note that shifts are created anonymously, so there is no direct link to the employee that will eventually be assigned to the shift.

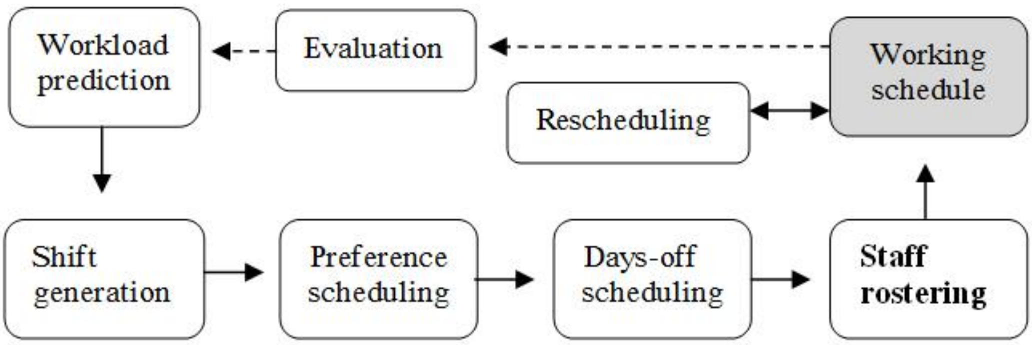

Fig. 1. The real-world workforce scheduling process.

In *preference scheduling*, each employee gives a list of preferences and attempts are made to fulfill them as well as possible. It is very important to pay attention to employee requests. Kellog and Walczak (2007) report that any academic nurse rostering model that does not include some opportunity for preference scheduling will probably not be implemented. Nurses tend to use complex decision-making skills when selecting their personal schedules. The employees' preferences are considered in the days-off scheduling and staff rostering phases.

Days-off scheduling deals with the assignment of rest days between working days over a given planning horizon. Days-off scheduling also includes the assignment of vacations and special days, such as union steward duties and training sessions. *Staff rostering*, also referred to as shift scheduling, deals with the assignment of employees to shifts. Days-off and shifts are often scheduled simultaneously. However, if a hospital scheduled days-off every tenth week and rostering staff every second week, the nurses would be able to plan their free time more conveniently.

Rescheduling deals with ad hoc changes that are necessary due to sick leaves or other no-shows. The changes are usually carried out manually using some level of computer support. Finally, participation in *evaluation* ranges from the individual employee through personnel managers (administrative nurses) to executives (head nurses). A reporting tool should provide performance measures in such a way that the personnel managers can easily evaluate both the realized staffing levels and the employee satisfaction. When necessary, the workload prediction and/or shift generation can be reprocessed and focused, and the whole workforce scheduling process restarted.

3 Nurse Rostering in an Intensive-Care Unit in a Finnish Hospital

We have experience in solving workforce scheduling problems occurring in the transportation industry, see for example (Nurmi and Kyngäs 2011, Nurmi et al. 2011, Kyngäs et al. 2012). Our current research is focused on workforce scheduling in call centers and hospitals. Based on our experiences, we believe that the framework for implementation-oriented staff scheduling we presented in (Ásgeirsson et al. 2011) can be used to model a considerable number of real-world workforce scheduling scenarios. With the help of administrative staff from Finnish hospitals we used the framework to describe the problem occurring in intensive-care units in Finnish hospitals. This problem includes five characteristics that are not always present in the nurse rostering cases reported in the academic literature:

1. The number of nurses is over 100
2. The nurses are grouped in four categories based on their total working hours within the planning horizon (100%, 78.43%, 50% and 40% of the full-time work)
3. Some shifts last more than 14 hours and actually include two consecutive shifts
4. Some nurses should always work on the same shifts
5. The nurses' wishes for days-off and shifts cover as much as 50% of their total work.

The implementation should present a wide variety of real-world constraints and be tractable enough to enable the addition of new constraints. It is important to concentrate on the acceptance by and satisfaction of both the administrative staff and the nurses. Despite the fact that the algorithm should be as robust as possible, no parameter tuning should be expected from the end-users. On the other hand, it should be possible for the end-users to influence different aspects of the algorithm, like weighting between constraints or limiting running times, if he/she wishes to.

We are well aware that it is difficult to incorporate the experience and expertise of the administrative nurses into a nurse rostering system. They often have extremely valuable knowledge, experience and detailed understanding of their specific staffing problem, which will vary from hospital to hospital. To formalize this knowledge into constraints is not an easy task. Still, we believe that the model given in this section builds up a solid foundation for nurse rostering scenarios in hospitals and specifically in intensive-care units.

The most important goal is to minimize understaffing and overstaffing. Low-quality rosters can lead either to an undersupply of nurses with a need to hire part-time nurses or an oversupply of nurses with too much idle time, implicating a loss of efficiency. The overall objective is to meet daily staffing requirements and personal preferences at minimum penalty without violating work contracts and government regulations. The framework presented in (Ásgeirsson et al. 2011) makes no strict distinction between hard and soft constraints; that will be given by the instances themselves. The goal in an instance is to find a feasible solution that is most acceptable for the hospital, that is, a solution that has no hard constraint violations and that minimizes the weighted sum of the soft constraint violations. The weights will also be given by the instances themselves and will vary between hospitals. Still, one should bear in mind that an instance is usually just an approximation of practice. In reality, hard constraints can turn out to be soft, if necessary, while giving weights to the soft constraints can be difficult.

The framework classifies the constraints into coverage, regulatory and operational requirements, and operational and personal preferences. The coverage requirement ensures that there are a sufficient number of nurses on duty at all times. The regulatory requirements ensure that the nurses' work contract and government regulations are respected. Operational and personal preferences should be met as far as possible; this leads to greater staff satisfaction and commitment, and reduces staff turnover.

We discuss the problem occurring in intensive-care units (ICU) in Finnish hospitals using an example from the Satakunta Hospital District which offers specialized medical care services for the 231,000 residents of the Satakunta region. The number of nurses in the ICU is 130. The problem can be modeled as follows; the constraint numbers refer to the constraints presented in (Ásgeirsson et al. 2011):

Coverage requirement

- (C1) An employee cannot be assigned to overlapping shifts
- (C2) A minimum number of employees with particular competences must be guaranteed for each shift
- (C4) A balanced number of surplus employees must be guaranteed in each working day

Regulatory requirements

- (R1) The required number of working days, working hours and days-off within a timeframe must be respected
- (R2) The required number of holidays within a timeframe must be respected
- (R3) The required number of free weekends (both Saturday and Sunday free) within a timeframe must be respected
- (R5) The minimum time gap of rest time between two shifts must be respected
- (R6) The number of special shifts (such as union steward duties and training sessions) for particular employees within a timeframe must be respected
- (R7) Employees cannot work consecutively for more than w days

Operational requirements

- (O1) An employee can only be assigned to a shift he/she has competence for
- (O2) At least g working days must be assigned between two separate days-off breaks
- (O5) An employee assigned to a shift type t_1 must not be assigned to a shift type t_2 on the following day (certain stints are not allowed)

Operational preferences

- (E1) Single days-off should be avoided
- (E2) Single working days should be avoided
- (E3) The maximum length of consecutive days-off is d
- (E4) A balanced assignment of single days-off and single working days must be guaranteed between the employees
- (E5) A balanced assignment of different shift types must be guaranteed between the employees
- (E7) A balanced assignment of weekdays must be guaranteed between employees
- (E8) Assign or avoid a given shift type before or after a free period (days-off, vacation)

Personal preferences

- (P1) Assign or avoid assigning given employees to the same shifts
- (P2) Assign a requested day-on or avoid a requested day-off
- (P3) Assign a requested shift or avoid an unwanted shift.

Often, a nurse cannot be assigned to more than one shift per day. However, two consecutive shifts per day are allowed in Finnish hospitals (see shift types C and E described later). The definition of constraint C1 allows two or more shifts to be assigned provided they do not overlap. Employees have seven possible competences: casting skill, intravenous skill, transportation skill, help skill, novice-nurse, intermediate-nurse and top-nurse. It is obvious that a nurse can only be assigned to a shift he/she has competence for (O1). The minimum number of employees of particular competences for time of day (C2) is given in Table 1. Note that the competences may overlap, e.g. a top nurse probably has an intravenous skill as well. Quite often, hospitals and ICUs have more nurses working than are needed to cover the minimum number of nurses each working day. The surplus nurses are used to cover the expected sick leaves and other no-shows. In our example case a balanced number of surplus nurses must be guaranteed in each working day (C4).

Table 1. Minimum number of employees with particular competences for each time of day.

	min #emp
Casting skill	1
Intravenous skill	11
Transportation skill	1
Help skill	1
Novice-nurse	0
Intermediate-nurse	5 (at night) 10 (otherwise)
Top-nurse	4

Within the last two years the hospital has started to consider the patient flow as a basis for the shift structure. Even though the shift structure is not near-optimal, it is a good start towards generating the shifts based on real workload prediction in the near future. The shift structure for the ICU is given in Table 2. Note that C and E are so-called double-shifts that last 14 hours and 30 minutes.

Table 2. The shift structure.

Code	Description	From	To
A	Morning	07.30	15.15
X	Transport I	07.30	15.15
U	Admin	07.30	15.30
B	Help I	07.30	16.00
C	Double	07.30	22:00
E	Transport D	07.30	22.00
O	Acute I	10.00	18.00
Z	Acute II	12.00	20.00
F	Special	14.00	22.00
I	Evening	15.00	22.00
P	Help II	15.00	23.00
J	Transport II	15.00	23.00
R	Help III	16.00	24.00
D	Acute III	17.00	24.00
Y	Night	21.30	07.45

The planning horizon is six weeks. The total working hours for each full-time nurse are 229 hours and 30 minutes (R1). The working hours can also be 180h, 114h 45min or 181h 30min if a nurse is on part-time pension or has small children. The holidays (R2) and special shifts (R6) are included in the working hours.

The working days and shifts are built up using the following rules. The number of free weekends within a timeframe must be at least two (R3). At least nine hours of rest are required between two shifts (R5). Nurses cannot work consecutively for more than nine days (R7). At least two working days must be assigned between two separate days-off (O2). Single days-off and single working days should be avoided (E1 and E2). The maximum length of consecutive days-off is four (E3). A balanced assignment of single days-off and single working days must be guaranteed between the employees (E4). A nurse assigned to a night shift (code Y) must not be assigned to an early shift (A,X,U,B,E,C) the following day (O5). Furthermore, a night/early shift should be avoided before/after a free period (E8).

Each six-week planning horizon is preceded with a phase where nurses express their wishes for days-off and shifts (P2 and P3). These wishes cover as much as 50% of their total work on average. This is why a balanced assignment of different shift types cannot be guaranteed between the employees as given in constraint E5. The same holds for balancing the assignment of weekdays (E7). A special request is that some nurses should always work on the same shifts because they travel together to work from the nearby cities (P1).

As per the nurse rostering problem classification given in (De Causmaecker and Vanden Berghe 2011), the problem could be classified as $ASBC|V3O|PX$.

The next section gives an outline of the PEAST algorithm that is used to solve the problem occurring in intensive-care units (ICU) in Finnish hospitals and especially in the Satakunta Hospital District. Section 5 presents our computational results.

4 The PEAST Algorithm

The usefulness of an algorithm depends on several criteria. The two most important are the quality of the generated solutions and the algorithmic power of the algorithm (i.e. its efficiency and effectiveness). Other important criteria include flexibility, extensibility and learning capabilities. We can steadily note that our PEAST algorithm (Kyngäs 2011) realizes these criteria. The acronym PEAST stems from the methods used as Population, Ejection, Annealing, Shuffling and Tabu. It has been used to solve real-world school timetabling problems (Nurmi and Kyngäs 2007), real-world sports scheduling problems (Kyngäs and Nurmi 2009) and real-world workforce scheduling problems (Kyngäs and Nurmi 2011).

The PEAST algorithm is a population-based local search method. As we know, the main difficulty for a local search is

1) to explore promising areas in the search space that is, to zoom-in to find local optimum solutions to a sufficient extent while at the same time
2) avoiding staying stuck in these areas for too long and
3) escaping from these local optima in a systematic way.

Population-based methods use a population of solutions in each iteration. The outcome of each iteration is also a population of solutions. Population-based methods are a good way to escape from local optima. The PEAST algorithm uses GHCM, the Greedy Hill-Climbing Mutation heuristic introduced in (Nurmi 1998) as its local search method. The outline of the algorithm is given in Figure 2 and the pseudo-code of the algorithm is given in Figure 3.

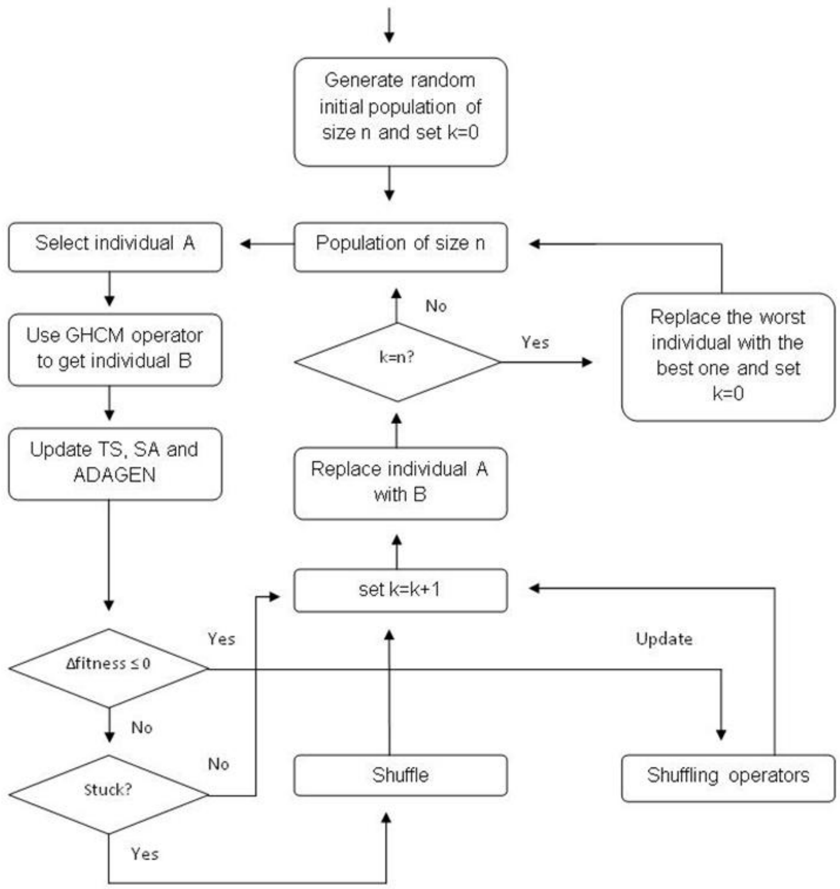

Fig. 2. The outline of the population-based PEAST algorithm.

The reproduction phase of the algorithm is, to a certain extent, based on steady-state reproduction: the new individual replaces the old one if it has a better or equal objective function value. Furthermore, the least fit is replaced with the best one when *n* better individuals have been found, where *n* is the size of the population. Marriage selection is used to select an individual from the population for a single GHCM operation. In the marriage selection we randomly pick an individual, *A,* and then we try at most $k-1$ times to randomly pick a better one. We choose the first better individual, or, if none is found, we choose *A*.

```
Set the time limit t, no_change limit m and the population size n
Generate a random initial population of individuals
Set no_change = 0 and better_found = 0
WHILE elapsed_time < t
    REPEAT n times
        Select an individual A by using a marriage selection with k = 3
        (explore promising areas in the search space)
        Apply GHCM to A to get a new individual A'
        Calculate the change Δ in objective function value
        IF Δ < = 0 THEN
            Replace A with A'
            IF Δ < 0 THEN
                better_found = better_found + 1
                no_change = 0
            END IF
        ELSE
            no_change = no_change + 1
        END IF
    END REPEAT
    IF better_found > n THEN
        Replace the worst individual with the best individual
        Set better_found = 0
    END IF
    IF no_change > m THEN
        (escape from the local optimum)
        Apply shuffling operators
        Set no_change = 0
    END IF
    (avoid staying stuck in the promising search areas too long)
    Update simulated annealing framework
    Update the dynamic weights of the hard constraints (ADAGEN)
END WHILE
Choose the best individual from the population
```

Fig. 3. The pseudo-code of the PEAST algorithm.

The heart of the GHCM heuristic is based on similar ideas to the Lin-Kernighan procedures (Lin and Kernighan 1973) and ejection chains (Glover 1992). The basic hill-climbing step is extended to generate a sequence of moves in one step, leading from one solution candidate to another. The GHCM heuristic moves an object, o_1, from its old position, p_1, to a new position, p_2, and then moves another object, o_2, from position p_2 to a new position, p_3, and so on, ending up with a sequence of moves.

Picture the positions as cells, as shown in Figure 4. The initial cell selection is random. The cell that receives an object is selected by considering all the possible cells and selecting the one that causes the least increase in the objective function when only considering the relocation cost. Then, another object from that cell is selected by considering all the objects in that cell and picking the one for which the removal causes the biggest decrease in the objective function when only considering the removal cost. Next, a new cell for that object is selected, and so on. The sequence of moves stops if the last move causes an increase in the objective function value and if the value is larger than that of the previous non-improving move. Then, a new sequence of moves is started. A tabu list prevents reverse order moves in the same sequence of moves. The initial solution is randomly generated.

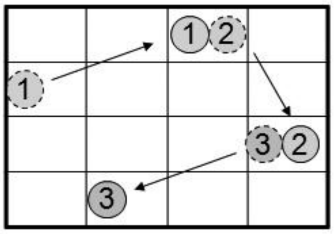

Fig. 4. A sequence of moves in the GHCM heuristic.

In the nurse rostering problem, each *row* corresponds to a nurse, and each *column* to a day. An *object* is a shift. A *move* involves removing a shift from a certain day and inserting it into another day.

The decision whether or not to commit to a sequence of moves in the GHCM heuristic is determined by a refinement (Nurmi 1998) of the standard simulated annealing method (Laarhoven and Aarts 1987). Simulated annealing is useful to avoid staying stuck in the promising search areas for too long. The initial temperature T_0 is calculated by

$$T_0 = 1 / \log(1/X_0) \,. \qquad \textbf{(1)}$$

where X_0 is the degree to which we want to accept an increase in the cost function (we use a value of 0.75). The exponential cooling scheme is used to decrement the temperature:

$$T_k = \alpha T_{k-1} \,, \qquad \textbf{(2)}$$

where α is usually chosen between 0.8 and 0.995. We stop the cooling at some predefined temperature. Therefore, after a certain number of iterations, *m,* we continue to accept an increase in the cost function with some constant probability, *p*. Using the initial temperature given above and the exponential cooling scheme, we can calculate the value:

$$\alpha = (-1/(T_0 \log p))^{-m} \,. \qquad \textbf{(3)}$$

We choose *m* equal to the maximum number of iterations with no improvement to the cost function and *p* equal to 0.0015.

A hyperheuristic (Cowling et al. 2000) is a mechanism that chooses a heuristic from a set of simple heuristics and applies it to the current solution, then chooses another heuristic and applies it, and continues this iterative cycle until the termination criterion is satisfied. We use the same idea, but the other way around. We apply shuffling operators to escape from the local optimum. We introduce a number of simple heuristics that are normally used to improve the current solution but, instead, we use them to shuffle the current solution - that is, we allow worse solution candidates to replace better ones in the current population. In the nurse rostering problem the PEAST algorithm uses two shuffling operations:

1) Move a random shift to a random day and repeat this l_1 times.
2) Swap two random shifts and repeat this l_2 times.

A random shuffling operation is selected every l/20th iteration of the algorithm, where l equals the maximum number of iterations with no improvement to the cost function. The best results were obtained using the values $l_1 = 5$ and $l_2 = 3$.

We use the weighted-sum approach for multi-objective optimization. A traditional penalty method assigns positive weights (penalties) to the soft constraints and sums the violation scores to the hard constraint values to get a single value to be optimized. We use ADAGEN, the ADAptive GENetic penalty method introduced in (Nurmi 1998) to assign dynamic weights to the hard constraints. This means that we are searching for a solution that minimizes the (penalty) function

$$\Sigma_i \alpha_i f_i(x) + \Sigma_i c_i g_i(x), \qquad \textbf{(4)}$$

where

α_i	= a dynamically adjusted weight for hard constraint i
$f_i(x)$	= cost of violations of hard constraint i
c_i	= a fixed weight for soft constraint i
$g_i(x)$	= cost of violations of soft constraint i

The hard constraint weights are updated every *k*th generation using the method given in (Nurmi 1998).

5 Computational Results

This section presents our results for solving a nurse rostering instance occurring in an intensive-care unit in the Satakunta Hospital District in Finland. The unit has 130 employees. Section 3 outlined the characteristics and constraints of the problem. Table 3 summarizes the hard and soft constraints of the problem. As the hard constraints state, the most important goal is to find a solution that has no overlapping shifts and guarantees a sufficient number of competences for each shift, and where employees do not work consecutively for more than nine days, have sufficient rest time between shifts and are not assigned to a forbidden shift before/after a night shift. As the soft constraint penalties state, the most important goal is to find individual rosters with exactly the required number of working hours. The rosters with less than 229 hours and 30 minutes for full-time nurses are considered as bad as the rosters with more than 229 hours and 30 minutes. The second most important goal is to fulfill the employee's requests.

Table 3 shows the manual solution and the PEAST solution to the problem. Neither solution has any hard constraint violations. The PEAST algorithm only needed 100 employees for generating a feasible and acceptable schedule. Note that the employees on vacation are not counted in this value. The PEAST algorithm was able to find a solution where all but one employee had exactly the required number of working hours. The algorithm also found a solution where 99% of all the employees' wishes were fulfilled even though those wishes covered as much as 50% of the employees' total work on average. Furthermore, the PEAST solution is clearly better at the number of single working days and finding a suitable weekend solution (see R3 and E7).

Table 3. The hard and soft constraints of the problem, the penalties for soft constraint violations, the manual solution and the solution obtained by the PEAST algorithm. The solutions indicate the number of violations for the constraints.

Constraint	Description	Penalty	Manual solution	PEAST solution
C1	Overlapping shifts	Hard	0 (108*)	0 (100*)
C2	Number of competences	Hard	0	0
C4	Balanced surplus employees	2	22	4
R1	Working hours / nurse	10	140	3
R3	Free weekends	4	50	16
R5	Sufficient rest time	Hard	0	0
R7	Consecutive working days	Hard	0	0
O1	Sufficient competence	Hard	0	0
O2	Working days in between	1	285	53
O5	Forbidden stints	Hard	0	0
E1	Single days-off	4	274	124
E2	Single working days	2	285	53
E3	Consecutive days-off	2	34	0
E4	Balanced singles	1	18	5
E5	Balanced shift types	1	37	46
E7	Balanced weekdays	1	343	127
E8	Forbidden shifts	4	0	0
P1	Same shifts	5	0	0
P2	Requested days-on	6	76% fulfilled	99% fulfilled
P3	Requested shifts	6	57% fulfilled	99% fulfilled

* The required number of employees needed for generating the schedule

The PEAST solution was found by generating ten solutions and selecting the best one. The algorithm was run on an Intel Core 2 Extreme QX9775 PC with a 3.2GHz processor and 4GB of random access memory running 64bit Windows Vista Business Edition. The best solution was found in 18 hours of computer time. The time may appear to be long. However, the point here is not to find a solution fast enough and with sufficient quality, but to find a solution of high quality. It is perfectly reasonable to run the algorithm overnight, because the solution is only needed once every six weeks. Note also that the manual solution took three weeks to generate. The detailed data for the instance can be obtained from the authors by email.

The Hospital Board members were very satisfied with our results. We are currently negotiating with them to optimize their overall workforce management process. This includes 1) generating an optimal shift structure based on the predicted patient flow, 2) optimizing the employees' preferences with a centralized self-scheduling system and 3) optimizing the days-off and shift assignments. As was stated in Section 1, the best action plan for real-world nurse rostering research is to cooperate with both a problem owner and a software vendor. We have a business partner that has workforce management software that already includes our optimization component. We are now looking to include nurse rostering in that software as well.

6 Conclusions and Future Work

We described an effective method for rostering nurses in an intensive-care unit in a Finnish hospital. The rosters were generated using a population-based local search method called the PEAST algorithm. The acronym PEAST stands for Population, Ejection, Annealing, Shuffling and Tabu, which represent the building blocks of the algorithm. The PEAST algorithm is flexible, easily extended and has good learning capabilities. The algorithm is based on a thorough local search method while still containing a strong global search element through the population based setup and randomized shuffling. The hospital was very satisfied with our results. We are currently negotiating with them to optimize their overall workforce management process. The PEAST algorithm has been integrated into market-leading workforce management software in Finland.

Future work includes modeling the instance presented in this paper using the xml-based modeling format introduced and managed by Tim Curtois (2010). We will also use the PEAST algorithm to solve the benchmark instances in (Curtois 2010). Our direction for future research is to strengthen our competence in workforce optimization concerning contact centers.

References

Alfares, H.K. (2004). Survey, categorization and comparison of recent tour scheduling literature. Annals of Operations Research 127, 145-175.

Ásgeirsson, E. I. (2010). Bridging the gap between self schedules and feasible schedules in staff scheduling. In Proc of the 8th Conference on the Practice and Theory of Automated Timetabling, Belfast, Ireland.

Ásgeirsson, E.I., Kyngäs, J. Nurmi, K., Stølevik, M. (2011). A Framework for Implementation-Oriented Staff Scheduling. Proceedings of the 5th Multidisciplinary International Scheduling Conference: Theory and Applications, Phoenix, USA.

Bard, J. F., Binici, C., Desilva, A. H. (2003). Staff Scheduling at the United States Postal Service. Computers & Operations Research 30, 745-771.

Bard, J., Purnomo H. (2005). Hospital-wide reactive scheduling of nurses with preference considerations. IIE Trans. 37(7), 589–608.

Bartholdi, J.J. (1981). A Guaranteed-Accuracy Round-off Algorithm for Cyclic Scheduling and Set Covering. Operations Research 29, 501–510.

Beddoe, G.R., Petrovic, S., Li, J. (2009). A Hybrid Metaheuristic Case-based Reasoning System for Nurse Rostering. Journal of Scheduling 12, 99–119.

Beer, A., Gaertner, J., Musliu, N., Schafhauser, W., Slany, W. (2008). Scheduling breaks in shift plans for call centers. In Proc. of the 7th Int. Conf. on the Practice and Theory of Automated Timetabling, Montréal, Canada.

Bilgin, B., De Causmaecker, P., Rossie, B., Vanden Berghe G. (2008). Local Search Neighbourhoods to Deal with a Novel Nurse Rostering Model. In Proc. of the 7th Int. Conf. on the Practice and Theory of Automated Timetabling, Montréal, Canada.

Burke, E., De Causmaecker P., Petrovic S., Vanden Berghe G. (2006). Metaheuristics for Handling Time Interval Coverage Constraints in Nurse Scheduling. Applied Artificial Intelligence, 743–766.

Burke, E.K., De Causmaecker, P., Vanden Berghe, G., Van Landeghem, H. (2004). The State of the Art of Nurse Rostering. Journal of Scheduling 7(6), 441–499

Chapados, N., Joliveau, M., Rousseau L-M. (2011). Retail Store Workforce Scheduling by Expected Operating Income Maximization, CPAIOR, 53–58.

Cowling, P., Kendall, G., Soubeiga, E. (2000). A hyperheuristic Approach to Scheduling a Sales Summit. Proceedings of the 3rd International Conference on the Practice and Theory of Automated Timetabling, 176–190.

Curtois, T. (Last update August 2010). Staff Rostering Benchmark Data Sets [Online]. Available: http://www.cs.nott.ac.uk/~tec/NRP/.

De Causmaecker, P., Vanden Berghe, G. (2011). A categorisation of nurse rostering problems. Journal of Scheduling, 14 (1), 3-16.

Diaz, T., Ferber, D., deSouza C., Moura A. (2003). Constructing nurse schedules at large hospitals. Internat. Trans. Oper. Res. 10(3), 245–265.

Dowling, D., Krishnamoorthy, M., Mackenzie, H., Sier, D. (1997). Staff rostering at a large international airport. Annals of Operations Research 72, 125-147.

Ernst, A. T., Jiang H., Krishnamoorthy, M., Sier, D. (2004). Staff scheduling and rostering: A review of applications, methods and models. European Journal of Operational Research 153 (1), 3–27.

Garey, M.R., Johnson, D.S. (1979). Computers and Intractability. A Guide to the Theory of NP-Completeness, Freeman.

Glover, F. (1992). New ejection chain and alternating path methods for traveling salesman problems. Computer Science and Operations Research: New Developments in Their Interfaces, edited by Sharda, Balci and Zenios, Elsevier, 449–509.

Kawanaka, H., Yoshikawa T., Shinogi T., Tsuruoka S. (2003). Constraints and search efficiency in nurse scheduling problem. In Proc. Internat. Sympos. Comput. Intelligence Robotics Automation 1, 312–317.

Kellogg D.L., Walczak S. (2007). Nurse Scheduling: From Academia to Implementation or Not. Interfaces 37(4), 355–369.

Kyngäs, J. (2011). Solving Challenging Real-World Scheduling Problems. Dissertation. Dept. of Information Technology, University of Turku, Finland.

Kyngäs, J., Nurmi, K. (2009). Scheduling the Finnish Major Ice Hockey League. Proceedings of the IEEE Symposium on Computational Intelligence in Scheduling, Nashville, USA.

Kyngäs, J., Nurmi, K. (2011). Shift Scheduling for a Large Haulage Company. Proceedings of the 2011 International Conference on Network and Computational Intelligence, Zhengzhou, China, 2011.

Kyngäs, N., Nurmi, K., Kyngäs, J. (2012). Optimizing Large-Scale Staff Rostering Instances. Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists, Hong Kong.

Lau, H. C. (1996). On the Complexity of Manpower Shift Scheduling. Computers and Operations Research 23(1), 93-102.

Lin, S., Kernighan, B. W. (1973). An effective heuristic for the traveling salesman problem. Operations Research 21, 498–516.

Lusby T., Dohn A., Range T., Larsen J. (2010). Ground Crew Rostering with Work Patterns at a Major European Airlines. In Proc of the 8th Conference on the Practice and Theory of Automated Timetabling, Belfast, Ireland.

Meisels, A., Schaerf, A. (2003). Modelling and solving employee timetabling problems. Annals of Mathematics and Artificial Intelligence 39, 41–59.

Meyer auf'm Hofe, H. (2001). Solving rostering tasks by generic methods for constraint optimization. Internat. J. Foundations Comput. Sci. 12(5), 671–693.

Nurmi, K. (1998). Genetic Algorithms for Timetabling and Traveling Salesman Problems. Dissertation. Dept. of Applied Math., University of Turku, Finland, 1998. Available: http://www.bit.spt.fi/cimmo.nurmi/

Nurmi, K., Kyngäs, J. (2007). A Framework for School Timetabling Problem. Proceedings of the 3rd Multidisciplinary International Scheduling Conference: Theory and Applications, Paris, France, 386–393.

Nurmi, K., Kyngäs, J. (2011). Days-off Scheduling for a Bus Transportation Staff. International Journal of Innovative Computing and Applications Volume 3 (1), Inderscience, UK..

Nurmi K., Kyngäs J., Post G. (2011). Driver Rostering for Bus Transit Companies. Engineering Letters 19(2), 125–132.

Stolletz, R. (2010). Operational workforce planning for check-in counters at airports. Transportation Research Part E 46, 414-425.

Tien J, Kamiyama A. (1982). On Manpower Scheduling Algorithms. SIAM Rev. 24 (3), 275–287.

van Laarhoven, P.J.M., Aarts, E.H.L. (1987). Simulated annealing: Theory and applications, Kluwer Academic Publishers.

Van Wezel, W., Jorna R. (1996). Scheduling in a generic perspective: Knowledge-based decision support by domain analysis and cognitive task analysis. Internat. J. Expert Systems 9(3), 357–381.