Pareto-Based Optimization for Multi-objective Nurse Scheduling

Edmund K. Burke, Jingpeng Li* and Rong Qu
{ekb, jpl, rxq}@cs.nott.ac.uk
School of Computer Science and Information Technology
The University of Nottingham
Nottingham, NG8 1BB, United Kingdom

* Corresponding author

Abstract: In this paper, we propose a search technique for nurse scheduling, which deals with it as a multi-objective problem. For each nurse, we first randomly generate a set of legal shift patterns which satisfy all shift-related hard constraints. We then employ an adaptive heuristic to quickly find a solution with the least number of violations on the coverage-related hard constraint, by assigning one of the available shift patterns to each nurse. Next, we apply a coverage repairing procedure to make the resulting solution feasible, by adding / removing any under-covered / over-covered shifts. Finally, to satisfy the soft constraints (or preferences), we present a simulated annealing based search method with the following two options: one with a weighted-sum evaluation function which encourages moves towards users' predefined preferences, and another one with a domination-based evaluation function which encourages moves towards a more diversified approximated Pareto set. Computational results demonstrate that the proposed technique is applicable to modern hospital environments.

1 Introduction

Nurse scheduling problems have been a research subject for a number of decades. We will briefly set the scene. For a comprehensive discussion of the various approaches that have been appeared in the literature, see the survey papers by Sitompul and Randhawa (1990), Cheang (2003) and Burke et al (2004b). Basically, the approaches range from traditional mathematical programming methods (Warner and Prawda, 1972; Beaumont, 1997; Jaumard et al, 1998; Bard and Purnomo, 2005) to special purpose heuristic methods (Isken and Hancock, 1990; Randhawa and Sitompul, 1993). One of the major research directions of nurse scheduling in recent years is the study of meta-heuristic methods, particularly evolutionary methods (Easton and Mansour, 1999; Aickelin and Dowsland, 2000; Kawanaka et al, 2001; Aickelin and Dowsland, 2004). Other meta-heuristics have also been investigated, including simulated annealing (Brusco and Jacobs, 1993; Thompson, 1996), tabu search (Dowsland, 1998; Burke et al, 1999), memetic algorithms (Burke et al, 2001), variable neighbourhood search (Burke et al, 2004a and 2007) and Bayesian optimization (Li and Aickelin, 2006). Many of these meta-heuristic approaches are attempting to solve models which capture the increasing complexity and wide range of demands required in modern hospital environments.

Nurse scheduling can be regarded as a type of resource allocation problem, in which the workload needs to be assigned to nurses periodically, taking into account a number of constraints and requirements. Hard constraints are those that must be satisfied in order to have a feasible schedule. They are often generated by physical resource restrictions and legislation. When requirements are desirable but not obligatory they are referred to as soft constraints, and are often used to evaluate the quality of feasible schedules. In nurse rostering, there are a large number of variations on legal regulations and individual preferences, depending on different countries and institutions. Typical issues concern coverage demand, day-off requirements, weekend-off requirements, minimum and maximum workforce (Burke et al, 2004b).

Hence, the nurse scheduling problem is inherently a multi-objective combinatorial problem, with each objective, possibly in conflict other objectives, corresponding to a soft constraint (or preference). However, until now there has been very limited work on the application of multi-objective techniques to this problem due to the complex nature of its real-world applications. Goal programming is the most commonly used method, which defines a target level for each criterion and relative priorities to achieve these goals, with the aim of finding a solution that is as close as possible to each of the objectives in the order of the priorities given (Arthur and Ravindran, 1981; Musa and Saxena, 1984; Ozkarahan and Bailey, 1988; Ozkarahan 1991; Chen and Yeung, 1993; Azaieza and Ai Sharif, 2005). Berrada et al (1996) proposed a tabu search, which considers only the most promising move to improve the objective function having the worst value at each iteration. Burke et al (2002) also presented a tabu search, but using the method of compromising programming to take all the objectives into account. Jaszkiewicz (1997) introduced a Pareto simulated annealing based on a weighted-sum objective function with adaptively changing weights, which is probably the first attempt to address the problem in terms of Pareto-based optimization.

In this paper, we present a Pareto-based search technique, towards the target of developing more flexible systems that are capable of addressing nurse scheduling problems in the real world. By applying an iterative heuristic which takes only the satisfaction of shift-related hard constraints into account, we randomly generate a set of legal shift patterns for each nurse. We then employ an adaptive heuristic to find a quick solution by assigning one of the available shift patterns to each nurse. However, solutions obtained at this stage are rarely feasible, let alone good quality, because the satisfaction of coverage demands (a hard constraint) has not been guaranteed because only a limited number of shift patterns generated and none of the soft constraints have been addressed yet. To deal with the coverage demands, we design a repairing heuristic which is capable of eliminating all the under-covers and over-covers within several iterations of its run. To satisfy the soft constraints associated with objectives, we proposed a simulated annealing based search method with two acceptance criteria to deal with the multiple objectives in different ways.

The paper is organized as follows. In Section 2, we introduce the nurse scheduling problem to be addressed. In Section 3, we formulate its integer programming model. In Section 4, we present an adaptive heuristic together with a coverage repairing procedure to obtain feasible solutions quickly. In Sections 5 and 6, we describe our proposed multi-objective simulated annealing approach and carry out the experiments on real world instances, respectively. We finally give our conclusions in Section 7.

2 The Nurse Scheduling Problem

The nurse rostering problem tackled here is based on the situation of intensive care units in a Dutch hospital, which involves the assignment of four types of shifts (i.e. shifts of *early*, *day*, *late* and *night*) within a planning period of 4-5 weeks to 16 nurses of different working contracts in a ward (Burke et al, 2007).

In brief, the problem has the following hard constraints:

- HC-1: Daily coverage demand of each shift type;
- HC-2: For each day, a nurse may start at most one shift;
- HC-3: Maximum number of working days of each nurse;
- HC-4: Maximum three on-duty weekends;
- HC-5: Maximum three *night* shifts;
- HC-6: No *night* shift between two non-*night* shifts;
- HC-7: Minimum two free days after a series of *night* shifts;
- HC-8: Maximum number of consecutive *night* shifts;
- HC-9: Maximum number of consecutive working days;
- HC-10: No *late* shifts for one particular nurse.

In addition, the problem has the following soft constraints:

- SC-1: Either no shifts or two shifts in weekends;
- SC-2: Avoiding a single day between two days off;
- SC-3: Minimum number of free days after a series of shifts;
- SC-4: Maximum number of consecutive assignments of a specific shift type;
- SC-5: Minimum number of consecutive assignments of a specific shift type;
- SC-6: Maximum number of weekly working days;
- SC-7: Minimum number of weekly working days;
- SC-8: Maximum number of consecutive working days for part-time nurses;
- SC-9: Avoiding certain shift type successions (e.g. *day* shift followed by *early* shift).

3 An Integer Programming Model

The above problem could be solved by a common approach of *Generation* and *Allocation*, which has been used to solve many personnel scheduling problems successfully (Fores et al, 2002; Li and Aickelin, 2004). The *Generation* phase first generates a large number of legal shift patterns (i.e. possible work patterns during the planning period) for each person, and the *Allocation* phase then allocates one of the shift patterns to each person to create a practical schedule. For our nurse scheduling problem, we use the following *Generation* steps to generate shift patterns for nurse *i*, $i \in I$ where *I* is the set of nurses.

Step 1 Let *J* be the set of days during the planning period, *W* be the set of weeks contained in the planning period, and *K* be the set of shift types including {1(*early*), 2(*day*), 3(*late*)}, 4(*night*)};

Step 2 Set index *g* = 1 and $A(i) = \phi$, where *A*(*i*) is the set of shift patterns of nurse *i*;

Step 3 Let A_{ig} be the *g*-th shift pattern generated for nurse *i*, represented as $A_{ig} = (a_{gjk} \mid \text{for } j = 1,...,|J| \text{ and } k = 1,...,|K|)$ where a_{gjk} is 1 if the *g*-th pattern covers shift type *k* on day *j* and 0 otherwise. Set a_{gjk} to be 0 or 1 at random, where 0 represents a free shift and 1 a working shift;

Step 4 Satisfy HC-2: if $(\sum_{k \in K} a_{gjk} > 1 \mid j \in J)$, randomly locate a $k' \in K$ having $a_{gjk'} = 1$. Set $a_{gjk} = 0 \mid \forall k \in K - \{k'\}$;

Step 5 Satisfy HC-4: if $(\sum_{w \in W} \sum_{k \in K} a_{g(7w)k} > 3)$, randomly locate a $w \in W$ having $a_{g(7w)k} = 1$. Set $a_{g(7w)k} = 0$;

Step 6 Satisfy HC-6: if $(a_{g(j-1)4} - a_{gj4} + a_{g(i+1)4} < 0 \mid j \in \{2,...,|J|-1\})$, randomly set either $a_{g(j-1)4} = 0$ or $a_{g(j+1)4} = 0$;

Step 7 Satisfy HC-8: if $(\sum_{j=r}^{r+n_1} a_{gj4} > n_1 \mid r \in \{1,...,|J|-n_1\})$ where n_1 is the maximum number of consecutive *night* shifts, randomly set either $a_{gr4} = 0$ or $a_{g(r+n_1)4} = 0$;

Step 8 Satisfy HC-5: if $(\sum_{j \in J} a_{gj4} > 3)$, randomly locate a $j \in J$ having $a_{gj4} = 1$. Set $a_{gj4} = 0$;

Step 9 Satisfy HC-7: if $(a_{gj4} + \sum_{k \in K} a_{g(j+1)k} + \sum_{k \in K} a_{g(j+2)k} \geq 2 \mid j \in \{2,...,|J|-1\})$, set $a_{g(j+1)k} = 0 \mid \forall k \in K$ and $a_{g(j+2)k} = 0 \mid \forall k \in K$;

Step 10 Satisfy HC-9: if $(\sum_{j=r}^{r+n_2} \sum_{k \in K} a_{gjk} > n_2 \mid r \in \{1,...,|J|-n_2\})$ where n_2 is the maximum number of consecutive working days, randomly locate a $j \in [r, r+n_2]$. Set

$a_{gjk} = 0 \mid \forall k \in K$;

Step 11 Satisfy HC-3: if $(\sum_{j \in J} \sum_{k \in K} a_{gjk} > m_i)$ where m_i is the maximum number of working days for nurse i, randomly locate a $j \in J$ having $\sum_{k \in K} a_{gjk} = 1$. Set $a_{gjk} = 0 \mid \forall k \in K$;

Step 12 Satisfy HC-10: if (i = 16), set $a_{gj3} = 1 \mid \forall j \in J$;

Step 13 Check the legality of A_{ig}. If illegal, go to step 3;

Step 14 Add A_{ig} to $A(i)$. If $(A_{ig} \notin A(i))$, set $g = g + 1$;

Step 15 If $g \le N(i)$ where $N(i)$ is the number of shift patterns to be generated for nurse i, go to step 3.

The *Allocation* phase can be modelled as the following Integer Programming (IP) problem. Decision variable x_{ig} is 1 if nurse i works on shift pattern g and 0 otherwise. Parameters J, K, $A(i)$ and a_{gjk} are defined in the same way as above. D_{jk} is the demand of nurses of shift type k on day j and c_{ig} is the preference cost of nurse i working on shift pattern g.

$$\text{Minimize } \sum_{i \in I} \sum_{g=1}^{|A(i)|} c_{ig} x_{ig} \tag{1}$$

Subject to:

$$\sum_{g=1}^{|A(i)|} x_{ig} = 1, \ \forall i \in I \tag{2}$$

$$\sum_{i=1}^{n} \sum_{g=1}^{|A(i)|} a_{gjk} x_{ig} = D_{jk}, \ \forall j \in J, k \in K \tag{3}$$

$$x_{ig} \in \{0,1\}, \ \forall i \in I, g \in \{1,...,|A(i)|\} \tag{4}$$

Objective function (1) minimizes the total cost of all nurses. Constraint (2) ensures that every nurse works exactly on one of his/her available shift patterns. Constraint (3) ensures that the demand for nurses of each shift type is fulfilled on every day, and constraint (4) ensures the integrality of variable x_{jg}.

By assigning every shift pattern the same cost value (e.g. $c_{ig} = 1 \mid \forall i, g)$, we have tried to solve the above IP problem by CPLEX 10.0, the latest version of a commercial IP solver. However, we could not find a single integer solution even by generating several million shift patterns for each nurse and allowing several days' runtime. Considering that the equality constraint in (3) might be too tight and that over-covered shifts which remain in a solution can be removed heuristically later, we relax (3) to be

$$\sum_{i \in I} \sum_{g=1}^{|A(i)|} a_{gjk} x_{ig} \ge D_{jk}, \ \forall j \in J, k \in K \tag{5}$$

Again, no integer solution can be found by using this approach regardless of the number of shift patterns generated and the maximum runtime allowed. This shows the complexity of the problem and proves that the traditional approach of *Generation* and *Allocation* is not sufficient to solve our problem alone. We are therefore looking for a more sophisticated technique to solve the problem effectively.

4 An Adaptive Heuristic with a Coverage Repairing Procedure

In this section, we present a heuristic search method to address the problem at the *Allocation* phase. It first carries out an Improved Squeaky Wheel Optimization (ISWO) search towards a solution with the least number of violations on coverage demands, and then applies a coverage repairing procedure to make the resulting solution feasible.

4.1 An Adaptive Heuristic Search Process by ISWO

The ISWO is based on the observation that the solutions of many real world problems consist of components which are intricately woven together in a non-linear, non-additive fashion. To deal with these components, Joslin and Clements (1999) proposed a technique called Squeaky Wheel Optimisation (SWO), and (Aickelin et al (2006) suggested an improved version called the ISWO which incorporates some evolutionary features, i.e. two additional steps of *Selection* and *Mutation*. In this section, we adapt the ISWO for our nurse scheduling problem. Starting from an initial solution, created by randomly assigning a shift pattern to each nurse, the steps of *Analysis*, *Selection*, *Mutation*, *Prioritization* and *Construction* are executed in a loop until a user specified parameter is reached or no improvement has been achieved for a certain number of iterations.

The first *Analysis* step evaluates the fitness of each component, i.e. a shift pattern assigned to each nurse, by taking the current schedule into account. The evaluation function used should be able to determine the contribution of this assignment towards the solution feasibility. Let S_i be the shift pattern assigned to nurse *i*. Its evaluation function can be formulated as

$$F(S_i) = \frac{\max(f_{1g_1},..., f_{ng_n}) - f_{ig_i}}{\max(f_{1g_1},..., f_{ng_n}) - \min(f_{1g_1},..., f_{ng_n})}, \ \forall i \in I \ , \tag{6}$$

where f_{ig_i} denotes the contribution of nurse *i* working on the g_i-th shift pattern towards reduction in nurse shortfall (as solving the problem of nurse shortfall is surely more important than that of nurse surplus). f_{ig_i} can be calculated as the number of shifts that would become uncovered if nurse *i* does not work his/her g_i-th shift pattern, formulated as

$$f_{ig_i} = \sum_{j \in J} \sum_{k \in K} a_{g_i jk} d_{jk}, \forall i \in I \ , \tag{7}$$

and

$$d_{jk} = \begin{cases} 1, & \text{if } (\sum_{i'=1}^{n} a_{g_{i'} jk} - a_{g_i jk} \le D_{jk}) \\ 0, & \text{otherwise} \end{cases}, \tag{8}$$

where a_{gjk} uses the same definitions as in formula (3), d_{jk} is 1 if there are still nurses of shift type *k* needed on day *j* before nurse *i* works on his/her assigned shift pattern and 0 otherwise.

The second *Selection* step determines whether a shift pattern S_i should be retained or discarded. The decision is made by comparing its fitness value $F(S_i)$ to a random number generated for each iteration in the range [0, 1]. If $F(S_i)$ is larger, then S_i will remain in its present allocation, otherwise S_i will be removed from the current schedule and the shifts it covers are then released with the coverage demands updated accordingly. By using the *Selection*, a shift pattern with a larger fitness value has a higher probability to survive in the current schedule.

The third *Mutation* step alters the shift patterns of the remaining nurses, i.e. it randomly discards them from the partial schedule at a small given rate p_m. The days and nights that a mutated S_i covers are then released and coverage demands are updated.

The fourth *Prioritization* step generates a new sequence for the nurses that are waiting to be rescheduled (i.e. the ones that have been removed by the steps of *Selection* and *Mutation*), with poorly-scheduled nurses being earlier in the sequence. Using the results of *Analysis*, this step first sorts the difficult (or removed) shift patterns in ascending order of their fitness values. As each shift pattern in the sequence is associated with a nurse, we can then obtain a sequence of difficult nurses.

The fifth *Construction* step repairs a schedule by assigning one of the shift patterns to each unscheduled nurse, in the order the nurses appear in the priority sequence. A new schedule is formed after each unscheduled nurse has been assigned a new shift pattern. We use the following constructing heuristic to schedule one nurse at a time to attempt to cover the largest number of uncovered shifts. For each shift pattern in a nurse's feasible set, it calculates the total number of uncovered shifts that would be covered if the nurse worked on that shift pattern. For instance, considering a short one-week planning period, we assume that a shift pattern covers Monday to Friday *day* shifts. We further assume that the current requirements for the *day* shifts from Monday to Sunday are as follows: (-4, 0, +1, -3, -1, -2, 0), where a negative number means undercover and a positive number means over-cover. Hence, the Monday to Friday *day*-shift pattern has a cover value of 8 as the sum of undercover is -8. If there is more than one shift pattern with the same highest undercover value, we choose the first one.

After each run of the above five steps, we need to calculate the fitness of the obtained solution so that the ISWO can always search from a best-improved solution. The chosen encoding of ISWO automatically satisfies constraints (2) and (4) of the IP formulation, and the target of ISWO is to achieve a solution that is as close to a feasible schedule as possible. Hence, in designing our fitness function, we can ignore the objective (1) and just evaluates the number of violations on coverage demands, i.e. try to satisfy constraint (3) as much as possible. We use the following function to calculate the fitness of an obtained solution:

$$\text{Minimize} \quad \sum_{j\in J}\sum_{k\in K}\left|D_{jk} - \sum_{i=1}^{n}\sum_{g=1}^{|A(i)|} a_{gjk}x_{ig}\right| \tag{9}$$

4.3 A Coverage Repairing Procedure

Due to the highly-constrained nature of the problem, solutions obtained by the ISWO are rarely feasible (i.e. there exists at least one under-covered or over-covered shift). To make the resulting solutions feasible, we employ a coverage repairing procedure to eliminate all the over-covers and under-covers as follows.

Step 1 Set initial solution $S = \{S_1, \ldots, S_n)$, where S_i denotes the shift pattern that nurse i works;

Step 2 Remove a single over-covered shift: if $(\sum_{i=1}^{n} a_{g_i jk} > D_{jk})$ and $(a_{g_i jk} = 1)$, set $a_{g_i jk} = 0$. Shift pattern S_i is thus revised. If S_i is infeasible which means at least one of the hard constraints has been violated, set $a_{g_i jk} = 1$;

Step 3 Remove two consecutive over-covered shifts: if $(\sum_{i=1}^{n}(a_{g_i jk} + a_{g_i(j+1)k}) > D_{jk})$ and $(a_{g_i jk} = a_{g_i(j+1)k} = 1)$, set $(a_{g_i jk} = a_{g_i(j+1)k} = 0)$. If the revised S_i is infeasible, set $(a_{g_i jk} = a_{g_i(j+1)k} = 1)$;

Step 4 Add a single under-covered shift: if $(\sum_{i=1}^{n} a_{g_i jk} < D_{jk})$ and $(a_{g_i jk} = 0)$, set $a_{g_i jk} = 1$. If the revised A_{ig_i} is infeasible, set $a_{g_i jk} = 0$;

Step 5 Add two consecutive under-covered shifts: if $(\sum_{i=1}^{n}(a_{g_i jk} + a_{g_i(j+1)k}) < D_{jk})$ and $(a_{g_i jk} = a_{g_i(j+1)k} = 0)$, set $(a_{g_i jk} = a_{g_i(j+1)k} = 1)$. If the revised A_{ig_i} is infeasible, set $(a_{g_i jk} = a_{g_i(j+1)k} = 0)$;

Step 6 Swap an under-covered shift with an over-covered shift: if $(\sum_{i=1}^{n} a_{g_i jk} < D_{jk})$ and $(a_{g_i jk} = 0)$ and $(\sum_{i'=1}^{n} a_{g_{i'} j'k'} > D_{j'k'})$ and $(a_{g_{i'} j'k'} = 1)$, set $a_{g_i jk} = 1$ and $a_{g_{i'} j'k'} = 0$. If one of the revised A_{ig_i} and $A_{ig_{i'}}$ is infeasible, set $a_{g_i jk} = 0$ and $a_{g_{i'} j'k'} = 1$;

Step 7 Check the feasibility of the revised schedule *S*: if $(\sum_{i=1}^{n} a_{g_i jk} \neq D_{jk} \mid j \in J, k \in K)$, go to step 2;

Step 8 Stop and output the final schedule *S*.

Providing a given schedule is infeasible but satisfies most of the coverage demands, the above coverage repairing procedure can transform this infeasible schedule to a feasible one quickly. However, under the rare circumstance where the procedure fails to make the repair, we should consider another initial schedule. This can be obtained by simply rerunning the ISWO for a different number of iterations, or from different sets of legal shift patterns.

5 A Multi-objective Simulated Annealing Approach for Nurse Scheduling

After obtaining a feasible solution which satisfies all the hard constraints by the above heuristic search method, we then need to edit it for practical use by satisfying the soft constraints as much as possible. Regarding the soft constraints, a hospital administrator normally has a general priority ordering in mind beforehand, but in making actual schedules such an ordering might not be implemented strictly. Hence, the constraint handling becomes a multi-objective problem, with each soft constraint associated with an individual goal. In this section, we present a simulated annealing approach to deal with the problem.

5.1 Objective Functions

We first define the decision variable x_{ijk} to be 1 if nurse *i* works on shift pattern *k* on day *j*, 0 otherwise. The definition of each parameter, if not specified separately, is the same as before. We use the following nine objective functions to formulate the corresponding nine goals.

1) Goal 1

This goal is to achieve complete weekends during the planning period, formulated as

$$\sum_{k \in K} (x_{i(7w-1)k} - x_{i(7w)k}) = 0, \forall i \in I, w \in W. \tag{10}$$

Thus, we can define the objective function $f_1(x)$ of goal 1 as

$$\text{Minimize} \quad f_1(x) = \sum_{i \in I} \sum_{w \in W} \left| \sum_{k \in K} (x_{i(7w-1)k} - x_{i(7w)k}) \right|. \tag{11}$$

2) Goal 2

This goal is to avoid any stand-alone shift during the planning period, formulated as

$$\sum_{k \in K} (x_{i(j-1)k} - x_{ijk} + x_{i(j+1)k}) \geq 0, \forall i \in I, j \in \{2, ..., |J|-1\}. \tag{12}$$

Thus, we can define the objective function $f_2(x)$ of goal 2 as

$$\text{Minimize} \quad f_2(x) = \sum_{i \in I} \sum_{j=2}^{|J|-1} \max \left\{ 0, \sum_{k \in K} (-x_{i(j-1)k} + x_{ijk} - x_{i(j+1)k}) \right\}. \tag{13}$$

3) Goal 3

This goal is to allocate at least two free days after a series of shifts during the planning period, formulated as

$$\sum_{k\in K}(x_{i(j-1)k} - x_{ijk} + x_{i(j+1)k}) \le 1, \forall i \in I, j \in \{2,...,|J|-1\}. \quad (14)$$

Thus, we can define the objective function $f_3(x)$ of goal 3 as

$$\text{Minimize} \quad f_3(x) = \sum_{i\in I}\sum_{j=2}^{|J|-1}\max\left\{0, \sum_{k\in K}(x_{i(j-1)k} - x_{ijk} + x_{i(j+1)k}) - 1\right\}. \quad (15)$$

4) Goal 4

This goal is to allocate at most a certain number of consecutive shifts of a particular shift type during the planning period, formulated as

$$\sum_{j=r}^{r+3} x_{ijk} \le c_k, \forall i \in I, r \in \{1,...,|J|-3\}, k \in \{1,3\}, \quad (16)$$

where c_k is the maximum number of consecutive shifts of type k. Thus, we can define the objective function $f_4(x)$ of goal 4 as

$$\text{Minimize} \quad f_4(x) = \sum_{i\in I}\sum_{r=1}^{|J|-3}\sum_{k\in\{1,3\}}\max\left\{0, \sum_{j=r}^{r+3} x_{ijk} - c_k\right\}. \quad (17)$$

5) Goal 5

This goal is to minimize the number of consecutive assignments of a specific shift type during the planning period, formulated as

$$x_{i(j-1)k} - x_{ijk} + x_{i(j+1)k} \ge 0, \forall i \in I, j \in \{2,...,|J|-1\}, k \in \{1,3\}. \quad (18)$$

Thus, we can define the objective function $f_5(x)$ of goal 5 as

$$\text{Minimize} \quad f_5(x) = \sum_{i\in I}\sum_{j=2}^{|J|-1}\sum_{k\in\{1,3\}}\max\{0, -x_{i(j-1)k} + x_{ijk} - x_{i(j+1)k}\}. \quad (19)$$

6) Goal 6

This goal is to allocate at most a certain number of weekly working days to each nurse with different working contract, formulated as

$$\sum_{j=7w-6}^{7w}\sum_{k\in K} x_{ijk} \le g_t, \forall t \in \{1,2,3\}, i \in I_t, w \in W, \quad (20)$$

where I_t is the subset of nurses working on the t-th contract satisfying I = {I_1 (full time), I_2 (short part time), I_3 (long part time)}, and g_t is the maximum number of weekly working days of nurses in subset I_t. Thus, we can define the objective function $f_6(x)$ of goal 6 as

$$\text{Minimize} \quad f_6(x) = \sum_{t=1}^{3}\sum_{i\in I_t}\sum_{w=1}^{|W|}\left[\max\left\{0, \sum_{j=7w-6}^{7w}\sum_{k\in K} x_{ijk} - g_t\right\}\right]. \quad (21)$$

7) Goal 7

This goal is to allocate at least a certain number of weekly working days to each nurse with different working contracts, formulated as

$$\sum_{j=7w-6}^{7w}\sum_{k\in K} x_{ijk} \ge h_t, \forall t \in \{1,2,3\}, i \in I_t, w \in W, \quad (22)$$

where h_t is the minimum number of weekly working days of nurses in subset I_t. Thus, we can define the objective function $f_7(x)$ of goal 7 as

$$\text{Minimize} \quad f_7(x) = \sum_{t=1}^{3}\sum_{i\in I_t}\sum_{w=1}^{|W|}\left[\max\left\{0, h_t - \sum_{j=7w-6}^{7w}\sum_{k\in K} x_{ijk}\right\}\right]. \quad (23)$$

8) Goal 8

This goal is to maximize the number of consecutive working days for part-time nurses during the planning period, formulated as

$$\sum_{j=r}^{r+3}\sum_{k\in K} x_{ijk} \le 3, \forall i \in I_1, r \in \{1,..., |J|-3\}\ . \quad (24)$$

Thus, we can define the objective function $f_8(x)$ of goal 8 as

$$\text{Minimize} \quad f_8(x) = \sum_{i\in I_1}\sum_{r=1}^{|J|-3} \max\left\{0, \sum_{j=r}^{r+3}\sum_{k\in K} x_{ijk} - 3\right\}. \quad (25)$$

9) Goal 9

This goal is to avoid certain shift type successions during the planning period, formulated as

$$x_{ijk_1} + x_{i(j+1)k_2} \le 2, \forall i \in I, j \in \{1,..., |J|-1\}, (k_1, k_2) \in K'\ , \quad (26)$$

where K' is the set of undesirable shift type successions including {(2,1), (3,1), (3,2), (1,4)}. Thus, we can define the objective function $f_9(x)$ of goal 9 as

$$\text{Minimize} \quad f_9(x) = \sum_{i\in I}\sum_{j=1}^{|J|-1}\sum_{(k_1,k_2)\in K'} \max\left\{0, x_{ijk_1} + x_{i(j+1)k_2} - 2\right\}. \quad (27)$$

5.2 Generation of Non-dominated Solutions by Simulated Annealing

With the above nine goals, the nurse scheduling problem can be regarded as a multi-objective optimization problem expressed as

$$f(x) = \text{Minimize } (f_1(x),..., f_9(x))\ . \quad (28)$$

The concept of dominance can be used to make a comparison between two solutions (Deb, 2005). A solution x is said to dominate another solution y if and only if $f_i(x) \le f_i(y)$ for i=1,…,9 and $f_i(x) < f_i(y)$ for at least one i. A solution is said to be globally non-dominated (or Pareto-optimal) if no other solution can dominate it. The set of all Pareto-optimal solutions is called the Pareto-optimal front (or the Pareto set), and solutions in the Pareto set represent the possible optimal trade-offs between conflicting objectives. A user can then select a preferred solution from the multi-objective set once it is revealed. When using (meta-)heuristic approaches, the non-dominated set produced will normally only be an approximation to the true Pareto front, thus in this paper we refer to the set generated by our approach as the archive of the approximated Pareto front.

Simulated annealing (SA) is a stochastic search algorithm first introduced by Kirkpatrick et al (1983) to a spin glass model. SA has been used to solve a wide variety of single objective optimization problems for more than twenty years. However, the applications of SA to multi-objective problems are very limited (Suman and Kumar, 2006). Most SA approaches still use the traditional weighted-sum objective functions. In this paper, we present a SA-based search method with two options to address user preferences in different ways: one weighted-sum evaluation function which encourages moves towards users' predefined preferences, and another domination-based evaluation function which encourages moves towards more non-dominated solutions which are well-spread in the approximated Pareto set.

We first define the neighbourhoods in which new solutions are generated. We apply the neighbourhoods of swapping blocks of consecutive shifts, which are inspired by the human scheduling process of re-allocating sections of schedules. Consecutive shifts within a period from one day to the whole planning period can be switched between any pair of two nurses in the schedule. To avoid violating the coverage demands again, swaps will only be made vertically. For a better illustration, we use Figure 1 to show the moves allowed in these neighbourhoods within a short 3-day planning period, with an arrow representing a possible move. Each day, a nurse can work at most one of the four shift types: *Early* (E), *Day* (D), *Late* (L) and *Night* (N).

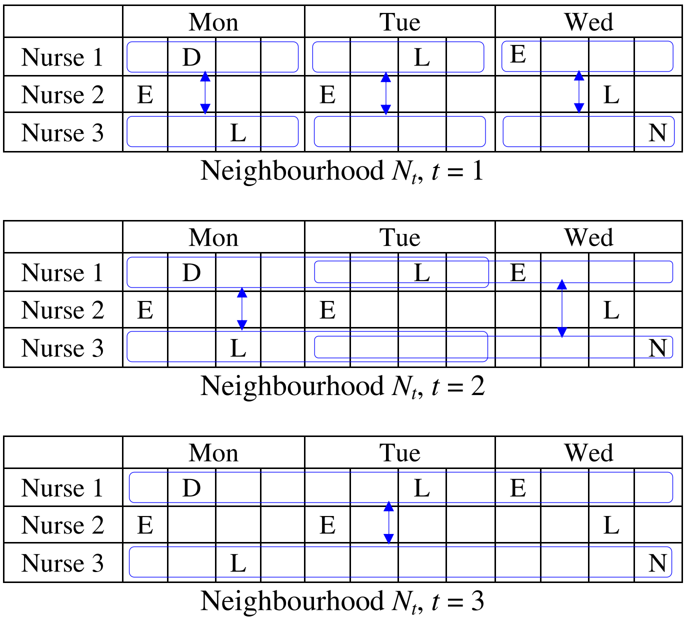

Figure 1. Possible moves in neighbourhoods N_t between nurse 1 and nurse 3

We then describe our proposed SA approach for multi-objective nurse scheduling as follows:

Step 1 Provide two options as the solution acceptance criteria of SA: option '1' for a weighted-sum evaluation function and option '2' for a domination-based evaluation function;

Step 2 Randomly generate a set of legal shift patterns for each nurse (see steps described in Section 3);

Step 3 Apply the ISWO to create a quick solution towards the least number of coverage violations (see Section 4.1), and then apply the coverage repairing procedure to obtain a feasible solution x (see Section 4.2);

Step 4 Let r be the number of runs. Set $r = 1$;

Step 5 Let k be the number of iterations. Set $k = 0$;

Step 6 Let $P(r)$ be the set of potentially non-dominated solutions. Set $P(r) = \{x\}$;

Step 7 Set current temperature $T(k)$ to an initial temperature T_0;

Step 8 Construct a new solution y by a random move within a randomly selected neighbourhood N_t of x;

Step 9 If y is infeasible, go to step 8;

Step 10 Replace x with y with acceptance probability $p_1 = \min(1, \exp(-\Delta E(y, x) / T))$: if the option is '1', $\Delta E(y,x) = \sum_{i=1}^{9} w_i(f_i(y) - f_i(x))$, where w_i is the priority weight of the i-th objective; if the option is '2', $\Delta E(y,x) = \|y\| - \|x\|$, where $\|y\|$ and $\|x\|$ denote the number of solutions in $P(r)$ dominating y and x respectively;

Step 11 If y is accepted and y is not dominated by x, update the set $P(r)$ with y in the following way: check y for Pareto dominance among all the solutions in $P(r)$, add y to $P(r)$ if it is non-dominated, and remove the solutions originally in $P(r)$ that are dominated by y.

Step 12 With a predefined small rate p_2, replace x with a randomly selected solution from set $P(r)$;

Step 13 Set $r = r+1$ and $k = k+1$. Decease $T(k)$ by using a proportional temperature cooling schedule: $T(k) = \alpha T(k-1)$, where cooling rate $\alpha \in [0.80, 0.99]$;

Step 14 Repeat steps 5-13 until a predefined number of iterations within SA is carried out;

Step 15 Repeat steps 2-14 until a predefined number of runs of SA is carried out.

Step 16 Set $P = P(1) + ... + P(r)$. Remove any solutions in P that are redundant or dominated by the other ones. Thus, P is our final set of Pareto-dominated solutions.

6 Computational Results

The proposed approach has been tested on a real-world problem with twelve data instances provided by ORTEC, an international consultancy company specializing in planning, optimization and decision support solutions. The hospital has a general preference ordering regarding the soft constraints listed in Section 2, which is {SC-1, SC-2} $\succ$ {SC-3} $\succ$ {SC-4, SC-5, SC-6, SC-7, SC-8} $\succ$ {SC-9}, where '$\succ$' denotes "*more preferred than*". However, due to the "soft" nature of these constraints, the above ordering is not necessarily the one that must be complied with strictly. While choosing a schedule for actual use, the hospital may consider candidate schedules with different trade-offs between the soft constraints, e.g. accept a schedule which just violates one or two constraints deemed as "highly preferred" in general, but satisfies all of the rest constraints. Hence, there is still a need for us to provide such a set of candidate schedules.

Three earlier approaches have been proposed on the same test instances. The first one is a hybrid genetic algorithm (Post and Veltman, 2004) which carries out a local search after each generation of the genetic algorithm to make improvement. The second is a hybrid Variable Neighbourhood Search (VNS) (Burke et al, 2007) which starts from an initial schedule created by an adaptive ordering technique, and sequentially runs the steps of VNS, feasibility correction, schedule disruption and schedule reconstruction in a loop until stopping criteria are met. The third is an IP-based VNS (Burke et al, 2006) which uses a non-shift-pattern-based IP to first solve a small problem including the full set of hard constraints and a subset of the soft constraints. It then executes a basic VNS to satisfy all the remaining constraints. The above three approaches solved the problem under a framework of single objective optimization. They all used the same weighted-sum objective function to combine all the objectives which can be outlined as follows:

$$\text{Minimize } f(x) = \sum_{i=1}^{9} w_i f_i(x) , \qquad (29)$$

where weights w_i were set to be (1000, 1000, 100, 10, 10, 10, 10, 10, 5) (Burke et al, 2007).

Table 1 lists the results of these three approaches after 1 hour's runtime. In general, the IP-based VNS has produced the best results. The hybrid genetic algorithm and the hybrid VNS were coded in Delphi 5 and implemented on a Pentium 1.7 GHz PC under Window 2000 operating system. The IP-based VNS was implemented on a 2.0 GHz PC under Windows XP, of which the IP part was solved by CPLEX 10.0 and the VNS part was coded in Java 2.

Data	Hybrid GA	Hybrid VNS	IP-based VNS
JAN	775	735	460
FEB	1791	1866	1526
MAR	2030	2010	1713
APR	612	457	391
MAY	2296	2161	2090
JUN	9466	9291	8826
JUL	781	481	425
AUG	4850	4880	3488
SEP	615	647	330
OCT	736	665	445
NOV	2126	2030	1613
DEC	625	520	405
AVE.	**2225**	**2145**	**1809**

Table 1. Results of the three earlier approaches after 1-hour runtime

Our SA-based approach is also coded in Java 2 and implemented on a 2.0 GHz PC under Windows XP. For each data instance, we allow the same maximum runtime of 1 hour. In addition, we set the number of shift patterns generated for each nurse to be 1000, the initial temperature of SA to be 100, replacement rate p_2 of SA to be 0.02, the cooling rate of SA to be 0.99 and the number of iterations within SA to be 1,000,000. Table 2 lists the results of using different evaluation functions on the obtained solutions. For the weighted-sum evaluation function, we use the same set of weight values as in formula (29), and list the number of archived non-dominated solutions (see column 2) and the best solution under this evaluations (see column 3). For comparison, we also list the relative percentage deviations of this best solution over the best solutions by the hybrid genetic algorithm (i.e. $\Delta\%^1$), the hybrid VNS (i.e. $\Delta\%^2$) and the IP-based VNS (i.e. $\Delta\%^3$).

Data	Weighted-sum objective function					Domination-based evaluation function (number of solutions)
	Number of solutions	Best solution	$\Delta\%^1$	$\Delta\%^2$	$\Delta\%^3$	
JAN	44	640	17.4	12.9	-39.1	1431
FEB	58	1645	8.15	11.8	-7.8	1744
MAR	39	1780	12.3	11.4	-3.9	976
APR	133	465	24.1	-1.8	-18.9	2235
MAY	48	1590	30.7	26.4	23.9	1154
JUN	27	9026	4.6	2.9	-2.3	876
JUL	56	446	42.9	7.3	-4.9	1362
AUG	76	1735	64.2	64.4	50.3	1865
SEP	90	339	44.9	47.6	-2.7	2033
OCT	85	540	26.6	18.8	-21.3	2287
NOV	39	1780	16.2	12.3	-10.4	1006
DEC	37	295	52.8	43.3	27.2	1192
AVE.	**61**	**1690**	**25.8%**	**21.5%**	**-0.8%**	**1513**

Table 2. Results of the our proposed multi-objective approach after 1-hour runtime

For the domination-based evaluation function, the comparison in the multi-objective test beds is difficult due to the lack of a systematic criterion to measure the performance of our Pareto-based approach. In multi-objective optimization, the objective value itself does not have a significant meaning. Rather, the configuration of objective values is more important. Hence, the commonly-used measure is only the plotting of the Pareto set. For a bi-objective problem, it is easy to draw a 2-D graph to show this measure. When the dimension of the objectives increases to three, it becomes harder to determine from a 3-D graph whether the Pareto set is a good one. When the dimension of objectives is larger than three, it is impossible to draw such a graph. Even if it were possible to plot the graph for more than three objectives, it would not be a good measure as no quantitative information exists. Therefore, for our nurse scheduling problem with nine objectives, apart from the total number of solutions in the Pareto set (see the last column in Table 2), we can only list a small number of example vector values for illustration purpose. Table 3 shows the details of such five solutions (i.e. columns #1, …, #5) for each data instance. For each column, it first lists the number of violations in terms of individual goals, and then lists the percentage of total number of violations in the solution with respect to the total number of constraints under consideration.

JAN	**#1**	**#2**	**#3**	**#4**	**#5**	**FEB**	**#1**	**#2**	**#3**	**#4**	**#5**
Goal 1	4	0	2	2	0	Goal 1	10	6	2	2	4
Goal 2	0	0	2	1	0	Goal 2	0	0	1	0	0
Goal 3	5	7	4	3	3	Goal 3	0	7	0	0	0
Goal 4	3	0	0	0	0	Goal 4	0	0	4	0	0
Goal 5	0	4	0	1	1	Goal 5	2	0	0	0	0
Goal 6	6	7	6	6	4	Goal 6	70	78	70	59	69

Goal 7	14	14	15	16	17	Goal 7	77	70	70	80	79
Goal 8	21	20	31	17	30	Goal 8	9	12	12	16	14
Goal 9	9	6	9	4	5	Goal 9	8	12	1	9	9
Sum(%)	0.7	0.6	0.8	0.5	0.7	Sum(%)	2.1	2.2	1.9	2.0	2.1
MAR	**#1**	**#2**	**#3**	**#4**	**#5**	**APR**	**#1**	**#2**	**#3**	**#4**	**#5**
Goal 1	2	2	0	0	2	Goal 1	4	6	8	10	8
Goal 2	0	1	0	0	1	Goal 2	0	0	0	0	0
Goal 3	4	2	3	2	6	Goal 3	3	2	1	3	2
Goal 4	3	0	0	5	0	Goal 4	0	0	0	5	0
Goal 5	0	10	5	0	5	Goal 5	1	6	1	0	0
Goal 6	70	73	70	68	65	Goal 6	7	2	5	5	3
Goal 7	72	70	73	74	73	Goal 7	4	4	5	4	4
Goal 8	13	23	33	19	9	Goal 8	12	17	16	10	11
Goal 9	7	0	0	1	4	Goal 9	7	2	5	8	7
Sum(%)	1.9	1.9	1.9	1.8	1.7	Sum(%)	0.4	0.4	0.5	0.5	0.4
MAY	**#1**	**#2**	**#3**	**#4**	**#5**	**JUN**	**#1**	**#2**	**#3**	**#4**	**#5**
Goal 1	8	12	12	6	4	Goal 1	19	13	11	7	7
Goal 2	0	0	0	0	0	Goal 2	0	0	0	0	0
Goal 3	6	5	2	1	0	Goal 3	6	4	2	3	3
Goal 4	0	0	3	0	2	Goal 4	0	0	2	4	0
Goal 5	3	12	4	3	0	Goal 5	6	1	0	5	1
Goal 6	46	17	44	39	36	Goal 6	130	100	100	90	90
Goal 7	24	24	30	30	24	Goal 7	55	75	74	86	87
Goal 8	20	39	12	32	34	Goal 8	13	9	27	26	18
Goal 9	17	15	9	6	8	Goal 9	2	3	3	13	12
Sum(%)	1.3	1.3	1.3	1.3	1.2	Sum(%)	2.6	2.3	2.5	2.6	2.5
JUL	**#1**	**#2**	**#3**	**#4**	**#5**	**AUG**	**#1**	**#2**	**#3**	**#4**	**#5**
Goal 1	2	8	10	4	8	Goal 1	4	4	2	0	2
Goal 2	0	0	0	0	0	Goal 2	0	0	0	0	0
Goal 3	5	3	5	3	4	Goal 3	5	9	8	9	1
Goal 4	0	0	0	1	0	Goal 4	0	3	0	3	0
Goal 5	3	0	0	0	3	Goal 5	4	3	4	5	4
Goal 6	3	2	2	2	2	Goal 6	70	60	65	64	66
Goal 7	4	4	4	5	4	Goal 7	49	44	41	40	40
Goal 8	25	21	8	21	9	Goal 8	15	20	14	11	39
Goal 9	5	4	5	3	2	Goal 9	4	9	6	7	5
Sum(%)	0.5	0.5	0.4	0.4	0.4	Sum(%)	1.6	1.6	1.5	1.5	1.7
SEP	**#1**	**#2**	**#3**	**#4**	**#5**	**OCT**	**#1**	**#2**	**#3**	**#4**	**#5**
Goal 1	2	2	6	4	10	Goal 1	8	6	0	6	4
Goal 2	1	0	0	0	0	Goal 2	0	0	0	0	0
Goal 3	3	0	5	4	2	Goal 3	5	3	6	2	2
Goal 4	0	0	1	1	1	Goal 4	0	0	1	0	1
Goal 5	3	2	1	0	0	Goal 5	4	0	2	3	0
Goal 6	0	2	0	3	4	Goal 6	8	7	8	9	7
Goal 7	2	3	3	0	3	Goal 7	14	15	14	15	20
Goal 8	11	18	8	7	8	Goal 8	17	25	16	17	12
Goal 9	1	2	6	6	0	Goal 9	5	4	3	2	10
Sum(%)	0.3	0.3	0.3	0.3	0.3	Sum(%)	0.7	0.7	0.6	0.6	0.6
NOV	**#1**	**#2**	**#3**	**#4**	**#5**	**DEC**	**#1**	**#2**	**#3**	**#4**	**#5**
Goal 1	2	0	8	2	2	Goal 1	4	0	0	4	6
Goal 2	0	0	0	0	0	Goal 2	0	0	0	0	0
Goal 3	5	4	2	5	3	Goal 3	1	5	5	2	3
Goal 4	0	0	1	2	0	Goal 4	0	4	0	0	1
Goal 5	3	0	1	2	4	Goal 5	0	4	4	1	0
Goal 6	100	90	91	88	81	Goal 6	0	3	8	6	1
Goal 7	55	48	50	51	60	Goal 7	4	4	2	0	2
Goal 8	4	14	15	17	13	Goal 8	9	7	3	6	5
Goal 9	4	10	4	4	1	Goal 9	4	0	2	4	5
Sum(%)	1.9	1.8	1.9	1.9	1.8	Sum(%)	0.2	0.3	0.3	0.3	0.3

Table 3. Example solutions found by the domination-based evaluation function

According to the results in Tables 2 and 3, we can see that our system with dual criteria of solution acceptance has provided a flexible way to well solve the nurse scheduling problem. By the aid of a weighted-sum evaluation function which combines users' explicit preferences on the objectives, our system can provide users with superior single solutions. For example, using the same objective function, our results are no worse than those of the IP-based VNS, and are significantly better than those of the hybrid genetic algorithm and the hybrid VNS (increased by 25.8% and 21.5% on average, respectively). In addition, by the aid of a domination-based evaluation function, our system can provide users with up to two thousand non-dominated solutions, some of which satisfy as high as 99.8% of the constraints. Hence, in the case that users do not have clear preferences on some specific objectives, they still may have plenty of choices in making the decisions. For example, to choose a schedule like the one shown in Figure 1.

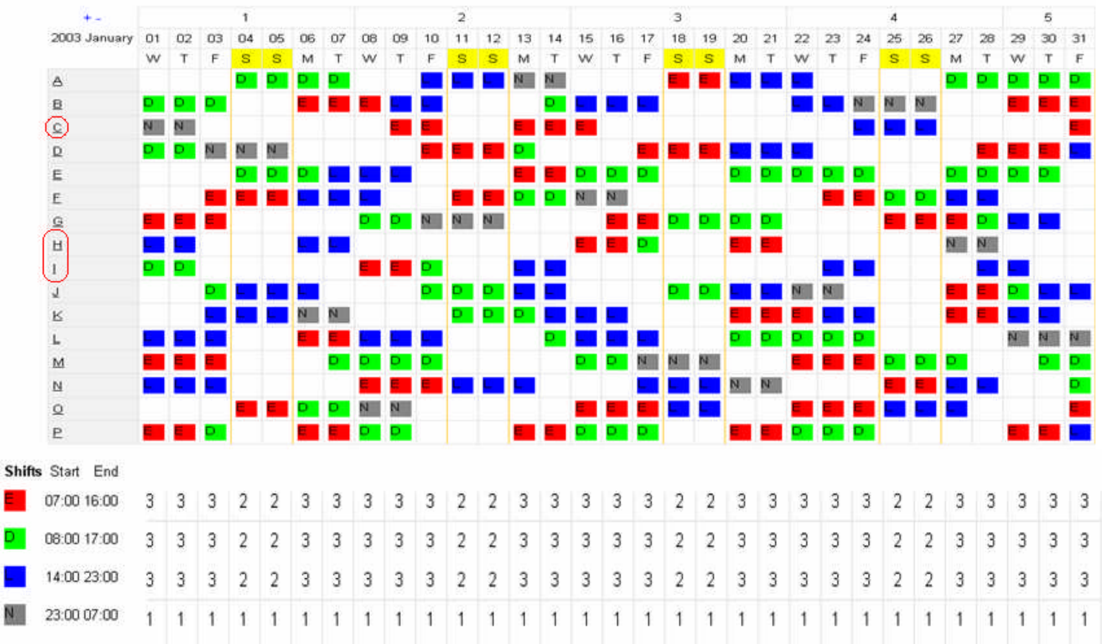

Figure 1. An example schedule for decision making

7 Conclusions

In this paper, we propose a Pareto-based search technique to solve the multi-objective nurse scheduling problem. We first design a generating heuristic which randomly generates a set of legal shift patterns for each nurse. We then employ an adaptive heuristic, called improved squeaky wheel optimization, to quickly find a solution with the least violations on coverage demands. Next, we apply a coverage repairing heuristic to make the resulting solution feasible. Finally, we propose a simulated annealing based search method with two options to address user preferences in different ways: one weighted-sum evaluation function which encourages moves towards users' predefined preferences, and another domination-based evaluation function which encourages moves towards a more diversified Pareto set.

The proposed approach has the following advantages. The first is its search ability which is demonstrated in a benchmark comparison by using the same weighted-sum evaluation function. The second is its adaptability as it can be applied to other hospital environments by simply altering the formulations of constraints and requirements. The third is its flexibility as it provides dual criteria of solution acceptance during the search, thus enabling users more degree of freedom for a better decision making.

Acknowledgements

The work was funded by the UK's Engineering and Physical Sciences Research Council (EPSRC), under grant GR/S31150/01. ORTEC kindly provided the test data.

References

[1] U. Aickelin and K. Dowsland, "Exploiting problem structure in a genetic algorithm approach to a nurse rostering problem," *Journal of Scheduling*, vol. 3, pp. 139-153, 2000.
[2] U. Aickelin and K. Dowsland, "An indirect genetic algorithm for a nurse scheduling problem," *Computers and Operations Research*, vol. 31, pp. 761-778, 2004.
[3] U. Aickelin, E.K. Burke, and J. Li, "Improved Squeaky Wheel Optimisation for Driver Scheduling," *Parallel Problem Solving from Nature*, Lecture Notes in Computer Science, vol. 4193, pp. 182-192, Springer, 2006.
[4] J. Arthur and A. Ravindran, "A Multiple Objective Nurse Scheduling Model," *IIE Transactions*, vol. 13, pp. 55-60, 1981.
[5] J. Bard and H.W. Purnomo, "Preference scheduling for nurses using column generation," *European Journal of Operational Research*, vol. 164, pp. 510-534, 2005.
[6] M.N. Azaieza and S.S. Al Sharifb, "A 0-1 goal programming model for nurse scheduling," *Computers & Operations Research*, vol. 32, pp. 491-507, 2005.
[7] I. Berrada., J.A. Ferland, and P. Michelon, "A multi-objective approach to nurse scheduling with both hard and soft constraints," *Socio-Economic Planning Science*, vol. 30, pp. 183-193, 1996.
[8] N. Beaumont, "Scheduling staff using mixed integer programming," *European Journal of Operational Research*, vol. 98, pp. 473-484, 1997.
[9] M.J. Brusco and L.W. Jacobs, "A simulated annealing approach to the cyclic staff-scheduling problem," *Naval Research Logistics*, vol. 40, pp. 69–84, 1993.
[10] E.K. Burke, P. De Causmaecker, and G. Vanden Berghe, "A hybrid tabu search algorithm for the nurse rostering problem," *Lecture Notes in Artificial Intelligence*, Springer, vol. 1585, pp. 187-194, 1999.
[11] E.K. Burke, P. De Causmaecker, S. Petrovic, and G. Vanden Berghe, "A Multi criteria meta-heuristic approach to nurse rostering," in *Proceedings of the 2002 Congress on Evolutionary Computation (CEC2002)*, pp. 1197-1202, 2002.
[12] E.K. Burke, P. De Causmaecker, S. Petrovic, and G. Vanden Berghe, "Variable neighborhood search for nurse rostering problems," in M.G.C. Resende and J.P. De Sousa (eds.), *Metaheuristics: Computer Decision-Making (Combinatorial Optimization Book Series)*, Kluwer, Chapter 7, pp. 153-172, 2004a.
[13] E.K. Burke, P. De Causmaecker, G. Vanden Berghe, and H. Landeghem, "The state of the art of nurse rostering," *Journal of Scheduling*, vol. 7, no. 6, pp. 441-499, 2004b.
[14] E.K. Burke, P. Cowling, P. De Causmaecker, and G. Vanden Berghe, "A memetic approach to the nurse rostering problem," *Applied Intelligence*, vol. 15, pp. 199–214, 2001.
[15] E.K. Burke, J. Li, and R. Qu, "A hybrid model of integer programming and variable neighbourhood search for highly-constrained nurse rostering problems," *Technique report*, no. NOTTCS-TR-2006-7, The University of Nottingham, 2006.
[16] E.K. Burke, T. Curtis, G. Post, R. Qu, and B. Veltman, "A hybrid heuristic ordering and variable neighbourhood search for the nurse rostering problem," *European Journal of Operational Research* (in print), 2007.
[17] B. Cheang, H. Li, A. Lim, and B. Rodrigues, "Nurse rostering problems – a bibliographic survey," *European Journal of Operational Research*, vol. 151, pp. 447-460, 2003.
[18] J.G. Chen, T. Yeung, "Hybrid expert system approach to nurse scheduling," *Computers in Nursing*, vol. 11, pp. 183-192, 1993.

[19] K. Deb, "Multi-objective optimization," In: E.K. Burke and G. Kendall (eds) *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Methodologies*, Chapter10, pp. 273-316, Springer.
[20] K.A. Dowsland, "Nurse scheduling with tabu search and strategic oscillation," *European Journal of Operational Research*, vol. 106, pp. 393–407, 1998.
[21] F.F. Easton and N. Mansour, "A distributed genetic algorithm for deterministic and stochastic labor scheduling problems," *European Journal of Operational Research*, vol. 118, pp. 505–523, 1999.
[22] S. Fores, L. Proll, and A. Wren, "TRACS II: a hybrid IP/heuristic driver scheduling system for public transport," *Journal of the OR Society*, vol. 53, pp. 1093-1100, 2002.
[23] H. Kawanaka, K. Yamamoto, T. Yoshikawa, T. Shinigi, and S. Tsuruoka, "Genetic algorithm with the constraints for nurse scheduling problem," in *Proceedings of Congress on Evolutionary Computation* (*CEC*), pp. 1123-1130, 2001.
[24] S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671-680, 1983.
[25] I. Isken and W. Hancock, "A heuristic approach to nurse scheduling in hospital units with non-stationary, urgent demand and a fixed staff size," *Journal of the Society for Health Systems*, vol. 2, pp. 24-41, 1990.
[26] A. Jaszkiewicz, "A metaheuristic approach to multiple objective nurse scheduling," *Foundations of Computing and Decision Sciences*, vol. 22, pp. 169-184, 1997.
[27] B. Jaumard, F. Semet, and T. Vovor, "A generalised linear programming model for nurse scheduling," *European Journal of Operational Research*, vol. 107, pp. 1-18, 1998.
[28] D.E. Joslin and D.P. Clements, "Squeak wheel optimisation," *Journal of Artificial Intelligence*, vol. 10, pp. 353-373, 1999.
[29] J. Li and U. Aickelin, "BOA for nurse scheduling," in: Pelican M, Sastry K and Cantú-Paz E. (eds), *Scalable Optimization via Probabilistic Modeling: From Algorithms to Applications*, Springer, Chapter 17, *pp*. 315-332, 1996.
[30] A. Musa and U. Saxena, "Scheduling nurses using goal-programming techniques," *IIE Transaction*, vol. 16, pp. 216-221, 1984.
[31] I. Ozkarahan, "An integrated nurse scheduling model," *Journal of the Society for Health Systems*, vol. 3, pp. 79-101, 1991.
[32] I. Ozkarahan and J.E. Bailey, "Goal programming model subsystem of a flexible nurse scheduling support system," *IIE Transaction*, vol. 16, pp. 306-316, 1988.
[33] G. Post and B. Veltman, "Harmonious personnel scheduling," in *Proceedings of the 5th International Conference on Practice and Automated Timetabling* (*PATAT*), pp. 557-559, 2004.
[34] S.U. Randhawa and D. Sitompul, "A heuristic based computerised nurse scheduling system," *Computers and Operations Research*, vol. 20, pp. 837-844, 1993.
[35] D. Sitompul and S. Randhawa, "Nurse scheduling models: a state-of-the-art review," *Journal of the Society of Health Systems*, vol. 2, pp. 62-72, 1990.
[36] B. Suman and P Kumar, "A survey of simulated annealing as a tool for single and multiobjective optimization," *Journal of the Operational Research Society*, vol. 57, pp. 1143-1160, 2006.
[37] G.M. Thompson, "A simulated annealing heuristic for shiftscheduling using non-continuously available employees," *Computers and Operations Research*, vol. 23, pp. 275–288, 1996.
[38] M. Warner and J. Prawda, "A mathematical programming model for scheduling nursing personnel in a hospital," *Management Science*, vol. 19, pp. 411–422, 1972.