

A Two-Stage Heuristic Approach for Nurse Scheduling Problem: A Case Study in an Emergency Department

Name of Author 1: T. C. WONG (corresponding author)

Department: Department of Design, Manufacture and Engineering Management

University/Institution: University of Strathclyde

Town/City: Glasgow, Scotland

Country: United Kingdom

Email: andywtc@graduate.hku.hk

Name of Author 1: M. XU

Department: Department of Systems Engineering and Engineering Management

University/Institution: City University of Hong Kong

Town/City: Hong Kong

Country: China

Email: maixu2@student.cityu.edu.hk

Name of Author 3: K. S. CHIN

Department: Department of Systems Engineering and Engineering Management

University/Institution: City University of Hong Kong

Town/City: Hong Kong

Country: China

Email: mekschin@cityu.edu.hk

ABSTRACT

Nurse scheduling is a critical issue in the management of emergency department. Under the intense work environment, it is imperative to make quality nurse schedules in a most cost and time effective way. To this end, a spreadsheet-based two-stage heuristic approach is proposed for the nurse scheduling problem (NSP) in a local emergency department. First, an initial schedule satisfying all hard constraints is generated by the simple shift assignment heuristic. Second, the sequential local search algorithm is employed to improve the initial schedules by considering soft constraints (nurse preferences) into account. The proposed approach is benchmarked with the existing approach and 0-1 programming. The contribution of this paper is twofold. First, it is one of a few studies in nurse scheduling literature using heuristic approach to generate nurse schedules based on Excel spreadsheet. Therefore, users with little knowledge on linear programming and computer sciences can operate and change the scheduling algorithms easily. Second, while most studies on nurse scheduling are situated in hospitals, this paper attempts to bridge the research gap by investigating the NSP in the emergency department where the scheduling rules are much more restrictive due to the intense and dynamic work environment. Overall, our approach generates satisfactory schedules with higher level of user-friendliness, efficiency, and flexibility of rescheduling as compared to both the existing approach and 0-1 programming.

Keywords: Nurse scheduling; Emergency department; Heuristics; Excel; 0-1 programming

1. Introduction

The healthcare expenditure in Hong Kong has reached HK\$84.4 billion in 2009, about 5.1% of the Hong Kong GDP and a projected increase to 9.2% by 2033. However, Hong Kong's total healthcare expenditure as a share of GDP is still far behind the world leading economies such as United Kingdom (9.8%), Australia (8.7%), Canada (11.4%), and United States (17.6%) in year-on-year comparison [1]. Due to the economic recession, an increasing number of experienced nurses have flowed to private healthcare institutes for higher salaries and lighter workload. This trend has led to a nurse shortage, especially the senior nurses in public hospitals. Emergency Department (ED) is the frontline of providing emergency care in a hospital, unfortunately it also suffers from the nurse shortage problem and has been under the criticism of declined service quality. At the meantime, there are more non-urgent patients seeking for ED services these years for the better facility and lower cost as compared to private clinics, and this further increases the workload of nurses. The problem mentioned can be boiled down to the mismatch between the increasing demands and the shrinking nursing manpower in the public healthcare institutes.

Nurse Scheduling Problem (NSP) is a critical part of ED resources management. A good schedule can contribute to improving nurse well-being and workload distribution, and hence enhance the work efficiency and patient experience. More important, it allows ED decision makers to better manage the scarce ED resources. Therefore, a good schedule can benefit both sides of patients and nurses. To a larger extent, NSP belongs to a broad topic of staff scheduling. This topic has been extensively studied by operational researchers and computer scientists for many years. According to Ernst et al. [2], staff scheduling can be defined as “the process of constructing work timetables for its staff so that an organization can satisfy the demand for its goods and services”. A good schedule should maximize the satisfaction of well-defined constraints such as government regulations, working practices, and personnel preferences, etc. A significant number of scheduling problems have been studied over the past few years in other commercial sectors, such as manufacturing [3-6], sales [7], airline crew [8-13], bus driver [14-16], and call center [17,18]. Unlike many commercial organizations, EDs provide around-the-clock services and the restrictive scheduling rules are very restrictive due to frequent manpower turnovers and unpredictable changes of shifts. Therefore, it may be difficult to apply the similar scheduling techniques that have been successfully used in other industries to solve the NSP in ED. Even the NSP in ED can be formulated in mathematical expressions without much difficulty, it is still very challenging to generate quality schedules using low-end computer tools that nurses can operate without extra knowledge.

In this study, we attempt to propose a heuristic approach to solve NSP in a local ED. Figure 1 illustrates an overview of the NSP which involves several phases from macro level planning to task rosters for each 8-hour shift and three shifts a day. In the planning phase, the nurse staffing level of three shifts is estimated based on daily patient visits and upper limit of

waiting times [19]. Other information such as the qualification/seniority of nurses, legal regulations, and nurse preferences on shifts have also been collected. In the scheduling phase, a two-stage heuristic approach combining simple shift assignment heuristic and sequential local search (SLS) is proposed, and it is benchmarked with 0-1 programming. Real-time rescheduling is very useful in ED environment where there are frequent changes of shifts/day-offs. Shift task rostering assigns primary ED duties such as triage, resuscitation, minor patient handling, etc. This paper focuses on the scheduling part.

[Figure 1 near here]

The organization of this paper is as follows: Section 2 provides an extensive literature of NSP studies in the last decade. Section 3 describes the nursing resources and existing scheduling approaches of the target ED. Section 4 presents the details of the proposed approach. Section 5 describes the formulation of 0-1 programming model. Section 6 reports the comparison between different approaches in terms of several measures. Section 7 gives concluding remarks and future work.

2. Literature Review

In general, there are four scheduling methods which have been widely used. They are cyclical scheduling, mathematical programming, heuristic scheduling, and artificial intelligence (AI). In cyclical scheduling, the shift and vacation assignment are fixed and repeated over certain time period. Hence, cyclical scheduling can minimize the shift uncertainty and allow nurses to be well-prepared for their assigned duties over a certain time period [20]. Marchionno [21] identified seven steps to help schedule the nurses in three-shift rotation. However, this approach is not suitable to the target ED in which the shifts/day-offs are often changed to accommodate unexpected urgent issues of nurses.

Mathematical programming approach is a traditional technique which has been widely applied to NSPs. This approach aims to search over a large solution space in order to locate the best solution such that the objective function can be optimized. Some previous studies have dedicated to mathematical programming approach, including integer programming [22-28], multi-objective goal programming [29-32], multicommodity flow [33]. Unlike other three approaches, mathematical programming approach guarantees optimal solutions. However, when the number of variables and constraints become large, the searching of optimal solutions would become very time-consuming. Therefore, in many recent studies, mathematical programming is integrated with other approaches such as GA [34] and local search approaches [32,35]. In this hybrid approach, the mathematical programming searches for the preliminary solutions satisfying all hard constraints, and then GA and local search approaches refine the solutions by maximizing the fulfilment of soft constraints. Such approach is able to find near-optimal solutions but consume much less computational time.

In heuristic scheduling, a set of scheduling rules are collected for constructing a decision-making tree which guides the generation of a feasible schedule under a variety of scenarios. These rules are formulated based on ED policies and regulations, service pledges, nurse preferences, etc. Some major heuristic techniques include knowledge-based techniques [36], adaptive heuristic [37], combinations of heuristic approaches [35,38], tabu and other local search [23,24,28,39-47], mix of tabu search and integer programming [23,32,35,48-50]. A typical example of local search technique is variable neighbourhood search (VNS) which is based on the idea of systematic change of neighbourhood area to identify better local optima [51]. It has been applied to a wide range of decision and optimization problems [32,35]. A comprehensive summary of VNS can be found in [52]. The motivation of developing heuristic approaches is to overcome the challenge of long computation time required by traditional mathematical programming approach, but the optimality of solutions cannot be guaranteed.

Many AI techniques have been applied to solve NSP, such as simulated annealing [31,53], ant colony optimization [54], and genetic algorithm (GA). Among them, GA is the most popular one. Cheang et al. [55] and Burke et al. [56] conducted extensive literature review on the application of GA since it was first introduced to NSP by Colomi et al. [57]. Unlike the local search techniques such as tabu search, GA can search the global near-optimal solutions in multiple points simultaneously so that it is less likely to be trapped into local optima. Some applications of GA to NSP can be easily found in the literature [e.g. 34,58-65]. Using GA approach, Maenhout and Vanhoucke [66] systematically compared various crossover operators applying to a standard NSP, and proposed several options of hybridizing multiple crossover operators. A novel attempt by Yeh and Lin [63] adopted the waiting time of ED patients as objective function and assessed a number of feasible solutions produced by GA. Although GA does not guarantee optimal solutions, its powerful searching ability makes it a popular approach for NSP. Also, by integrating other local search algorithms, the schedule quality can be further improved.

Overall, three points can be summarized from the extensive literature review. First, most NSPs are situated in hospitals. The scheduling rules in EDs could be more complicated and very few papers have investigated that [e.g. 36,63,64]. Second, since the NSPs are different in many aspects, it is difficult to draw conclusion on the best approach. Third, most of the previous studies only handle simplified or standard NSPs, and very few of them are eventually verified under real ED settings.

3. Case Description

The NSP in this study is situated in a local ED where the daily patient visits normally fall between 350 and 420. Due to the manpower shortage, nurses have been working under great pressure to meet stringent service targets such as patient waiting times for consultation. Therefore, it is imperative to develop quality nurse schedules which can meet varying patient demands and maintain a good life-work balance. The following subsections describe the

details of this NSP.

3.1 Nurses, Shifts, and Day-offs

In this ED, the nurses are classified in terms of gender, qualification (nurse officer and registered nurse), and experiences (junior, medium, and senior). There are seven types of nurses in total (Table I). The total number of nurses available to be scheduled varies between 50 and 60 due to dynamic manpower turnover. For a typical crew of 55 nurses, a nurse officer needs to schedule about $55 \times 7 \text{ days} = 385 \text{ shifts/day-offs}$ for a new scheduling week. In real practice, a good staffing should be a proper mix of manpower with different skill sets and proficiency levels in order to handle different clinical scenarios. For example, triage needs to be done by nurse officers, resuscitation is usually operated by junior nurses, and only male nurses are allowed to conduct physical check-ups on male patients. The shifts and day-offs are assigned in comply with government regulations and ED policies as shown in Table II.

[Table I near here]

[Table II near here]

3.2 Existing Approach of Scheduling

In this ED, the nurse schedules are generated by nurse officers on a rotational basis. The challenge faced by nurse officers is the complexity of scheduling a large nursing crew meeting all the legal and ED-level constraints. Limited by the existing (manual) approach, the nurse preferences and well-being cannot be always taken into account. Moreover, it takes an average of 2.5-3 hours for a nurse officer to generate a feasible schedule, or at least 2 hours for a skilled nurse officer. Considering the key managerial role nurse officers play in ED, a less tedious scheduling process means a higher level of medical involvement in the ED, and that is critical to maintain service quality especially during peak times.

3.3 Constraint and Importance Weight

The constraints of making nurse schedule can be categorized into hard and soft groups. Hard constraints usually refer to the legal and ED regulation while soft constraints denote nurse preference on shifts which directly link to the physical and mental welfare [30]. A high quality schedule should meet all hard constraints and as many soft constraints as possible.

Importance weight is assigned to each soft constraint to penalize the violation. The value of weight should be reflective of the desirability of the corresponding soft constraint. Nominal group technique (NGT) is used here to determine the importance weights. NGT is a structured group decision-making process widely used in both industry and academia as a tool to generate ideas or solutions [67]. The basic steps of NGT can be depicted as follows [68]:

- Provide meeting for participants

- Identify the problem to be solved
- Silent generation of ideas/solutions in writing
- Round robin sharing of ideas/solutions
- Voting the ideas/solutions

In the target ED, NGT was applied to a group of 5 nurses (1 nurse officer, 1 male nurse, 3 female nurses). The group was made up in accordance to the size of different types of nurses described in Table I so that the results of NGT could be representative of the entire nursing crew. The participating nurses were asked to list out desirable preferences in an informal meeting. The results were then discussed and the top preferences were shortlisted for rating. Each nurse rated the importance of each preference on a 10-point scale where a higher point represents a more desirable preference. Nurses could choose to rate all preferences or just some of them. All the ratings given by a nurse must sum up to 10. The reason of using rating method instead of consecutive rankings is that the former doesn't implicate the linearity between levels of desirability, so that it can better describe the preferences. Table III presents ratings made by 5 nurses on 8 preferences. It is noted that the first four preferences were rated almost equally important while the rest were less so. To guarantee the satisfaction of the most important 4 preferences, the scheduling model will treat them as hard constraints while the rest as soft constraints. Table IV presents the full list of current hard constraints and Table V shows the soft constraints. Two types of constraints are imposed on the two dimensions of the scheduling table (Figure 2).

[Table III near here]

[Table IV near here]

[Table V near here]

[Figure 2 near here]

We must emphasize that the values of importance weight are only used to indicate the importance of nurse preferences as a projection of the group experience. The weight is highly case-dependent and its determination is largely subject to researcher's experience. Therefore, the values of weights here should not be deemed universal among different EDs and they may deviate from the true values which cannot be measured accurately. Given these facts, it is necessary to test the schedule robustness by proving that the schedule quality is not statistically associated with the weight values. For this reason, a sensitivity analysis is performed by assessing the schedule quality with different weight values. The details will be introduced in Section 6.

3.4 Quality Factors of Nurse Schedule

In this section, the five quality factors proposed by Oldenkamp [69] are discussed together

with the corresponding scheduling constraints in Tables IV and V. Since these five factors are independent to each other, any change measured by one quality factor does not affect others. They are described as below:

The *completeness* factor represents the fulfilment of staffing per shift. Shift coverage constraint $H_{4.1}$ accounts for this factor.

The *optimality* factor represents the degree to which nursing expertise is distributed over different shifts. Shift coverage constraints $H_{4.2}$ – $H_{4.4}$ account for this factor.

The *proportionality* factor represents the degree to which each nurse is assigned the same number of day shifts, night shifts and weekend day-offs over the scheduling period. In this study, the number of evening and night shifts are restricted to each nurse, and day-offs are prioritized to be assigned to weekends if they are not yet occupied by working shifts. The details will be given in Section 4.

The *healthiness* factor represents the degree to which the physical and mental welfare of the nurses are fulfilled. Preference constraints $H_{3.1}$ – $H_{3.4}$ and all soft constraints S_1 – S_4 account for this factor.

The *continuity* factor represents the degree to which the nurses are assigned to the same shift across consecutive days. However, from the results of nominal group meeting in Section 3.3, this factor is deemed as insignificant. Specifically, three consecutive evening shifts (3pm-11pm) are unanimously disapproved by the participants since family gathering at evening time is essential.

Overall, all the constraints can be associated with these five quality factors. Although the last factor is not agreed by the nurses in the target ED, it is believed that the hard and soft constraints adopted in this study well define the schedule quality.

4. The Proposed Approach

To improve the generation of nurse schedule under an intense healthcare environment in the target ED, an efficient approach combining shift assignment heuristic and sequential local search (SLS) is proposed. This two-stage approach can be operated in Excel spreadsheet using VBA. Figure 3 illustrates the mechanism of our proposed approach. In brief, a schedule is manually initialized by first confirming approve-in-principle (AIP) day-offs (e.g. maternity leave, compassionate leave, clinical training class, etc.) and H/AO day-offs requested in advance by the nurses. After that, the shift assignment heuristic is applied to assign shifts and day-offs to the unoccupied timeslots for all nurses in the schedule following the sequence $N \rightarrow E \rightarrow D \rightarrow O \rightarrow AO/H$ from Monday to Sunday. In case of any change request in the middle of week, the shift heuristic can be used to reschedule from that to-be-changed weekday to Sunday while keeping other weekdays unchanged. The shift assignment heuristic is executed iteratively until an initial schedule satisfying all hard constraints is obtained. In the second stage, SLS is used to refine the initial schedule by taking the soft constraints into account. Handling hard and soft constraints in separate steps allows a more efficient solution searching

due to the limited computing power of VBA. The objective function of the search is simply a weighted sum of violations of four soft constraints as below.

$$Z_{proposed} = w_1 N_{S1} + w_2 N_{S2} + w_3 N_{S3} + w_4 N_{S4}, \quad (1)$$

where N_{Sx} and w_x denotes number of violations and importance weight of the corresponding soft constraint. For the ease of presentation, the importance weight of all soft constraints is doubled so that the maximum weight becomes 10 (Table V). This scaling is used in calculating the objective function values in Section 6.

[Figure 3 near here]

The operation of a complete two-stage algorithm is referred to an *iteration*. Multiple *iterations* are needed to reach a pre-defined target of objective function value. The operation of running multiple *iterations* to generate a satisfactory schedule based on an initial scenario (H_1) is referred to a *trial*. In the following subsections, details of two stages will be illustrated with a real NSP.

4.1 Shift Assignment Heuristic

As mentioned in Section 4, the nurse officer needs to first manually assign the AIP and H/AO day-offs to the corresponding nurses before shift assignment heuristic is executed. Based on the cumulative count of the unused H and AO day-offs, nurses with more than 3 unused H/AO day-offs are mandated to take these day-offs in the scheduling week. In Figure 2, shifts and day-offs of Sunday and AIP day-offs pre-assigned for the scheduling of next week are highlighted in bold. Including Sunday of the last scheduling week into the current one is to maintain the constraint continuity between two scheduling weeks. After that, shift assignment heuristic is executed to generate an initial schedule. The algorithm starts by randomly assigning the shifts and day-offs for all nurses on Monday by following the column constraints (i.e. H_{4-1} to H_{4-4}). The algorithm keeps going to fill the columns from Tuesday to Sunday while backtracking if the row constraints (i.e. H_{2-2} , H_{2-3} , and H_{3-1} to H_{3-4}) of each nurse are violated. Different shifts and day-offs are assigned separately following a certain sequence. In the current manual scheduling, nurse officers follow the sequence of $N \rightarrow E \rightarrow D \rightarrow O \rightarrow AO/H$. Three working shifts are always fixed before day-offs because they are associated with more complicated hard constraints. N is assigned in front of E and D because it has much lower staffing level. However, no experiments have been done to show which one between E and D should be assigned first. Decision criteria concerning this issue should include the calculating time and the objective function value of the resulting initial schedule. A side-by-side comparison between two sequence $N \rightarrow E \rightarrow D$ and $N \rightarrow D \rightarrow E$ will be made in the Section 6, and whichever the better in terms of the two criteria will be used for

further analysis. After the assignment of N, E, D, and O shifts, if any nurse hasn't been fully occupied over the week, AO or H day-offs are lastly assigned to complete the schedule. The AO or H day-off, whichever has larger counts (updated after initialization step), is assigned first. The shift assignment heuristic runs iteratively until a feasible solution satisfying all hard constraints is found. The algorithm of shift assignment heuristic is presented in Appendix I-A. Next subsection will introduce how SLS is applied to meet the soft constraints.

4.2 Sequential Local Search (SLS)

In the second stage, SLS is proposed to resolve soft constraint violations based on the initial schedule from the first stage. It handles the violations from the highest- to lowest- weighted soft constraint separately so that the most desirable soft constraint can be satisfied first. In this study, SLS is performed following the sequence of $S_2 \rightarrow S_3 \rightarrow (S_1, S_4)$.

Unlike many other local/neighbourhood search methods, SLS algorithm does not perform exhaustive search given the computational limitation of VBA. The SLS algorithm works on the “check, swap, repair (if necessary)” mode. For each soft constraint in the sequence, it first checks if there are violations nurse by nurse in the horizontal direction (Figure 2). If a *violation area* is identified, certain SLS operators will perform swaps between shifts/day-offs to correct the violation. Table VI shows the four soft constraints and the associated SLS operators. For each constraint, the best *swap option* is the one producing the largest objective function value reduction. The more *iterations* performed, the better the final schedule would be. As SLS allows violation of a lower-weighted constraint to correct violation of a higher-weighted constraint, it can be expected that many low-weighted violation will backlog in the late phase of an *iteration*, such as S_1 . These violations are extremely difficult to be repaired because any SLS operation can easily trigger new higher-weighted violations. This problem will be discussed in Section 6.

The algorithm of SLS is presented in Appendix I-B. Three SLS operators are described in the following subsections with illustrations provided in Appendix II, where the *violation areas* are shaded and the *swap positions* are bolded.

Vertical swap (Appendix II-A): A shift/day-off violating any soft constraints is swapped with another shift/day-off within the same column (same day). A swap is performed only if it does not trigger new violations of hard constraints or higher-weighted soft constraints. If multiple swaps are available, only the one resulting in maximum reduction of objective function value is performed.

Double swap_1 (Appendix II-B): This operator performs swaps in two steps. In the first step, a shift/day-off violating any soft constraint is swapped with another shift/day-off within the same row (same nurse). The operator searches for a swappable shift/day-off from the both side of the violating shift/day-off in horizontal direction such that a valid swap can be performed (e.g. E of Nurse_5 on Sun is swapped with O of Nurse_5 on Tue). If there are more than one swap options, the operator will choose one which causes no new violations or a new

violation with smaller weight value. When multiple best swap options are available, a swap is chosen randomly. If the best swap option triggers a new violation with the same or larger weight value, this swap is then aborted and the current violation is marked as temporarily *unsolvable*. Obviously, a swap should be also aborted if it violates any hard constraints. If the first step results in violation of shift coverage constraints ($H_{4.1}$ – $H_{4.4}$) in any of the two columns where the swapped shifts/day-offs are located, the second step is needed to repair the infeasible solution. The repairing swaps take place only in vertical direction in these two columns. If there are more than one swap options, the same decision criteria are used as in the first step.

Double swap_2 (Appendix II-C): This operator is the same as *double swap_1* except that the swaps take place only within the *violation area*.

[Table VI near here]

The motivation of investigating SLS in the second stage is in the consideration of limited computational power of VBA and the harsh constraints imposed on the shift coverage ($H_{4.1}$ – $H_{4.4}$). First, although SLS does not guarantee optimal solutions, it is able to reach near-optimum with less computational effort, just like many local/neighbourhood research techniques introduced in Section 2. If GA is used instead, operating GA on Excel spreadsheet would be a prohibitive computational burden to VBA, and the offspring schedules generated from crossovers and mutations are difficult to fulfil the hard constraints in two dimensions. Burke et al. [32] used VNS to research solutions through swaps between neighbourhoods in vertical direction only. This method is applicable in cases where all the nurses are treated as equals such that the daily shift coverage constraints will easily be fulfilled. However, VNS is not a feasible method in this study because the shift coverage in the target ED is a strict combination of gender, qualification, and seniority ($H_{4.1}$ – $H_{4.4}$).

5. 0-1 Programming Approach

To provide benchmark, a 0-1 programming model is formulated over the same scheduling period. A typical application of 0-1 programming to NSP was done by Azaiez and Sharif [30]. They used a subgrouping technique to split nurses, and for each subgroup the hard constraints were solved by LINGO independently. By aggregating the result of each subgroup, a complete schedule would be determined with satisfaction of all hard constraints. The main reason for nurse subgrouping is that it would be computationally exhaustive to generate an optimal schedule for a large group of nurses. However, if nurse subgrouping is applied in this study, the final schedule may easily violate one of the shift coverage constraints ($H_{4.1}$) since the staffing level of nurses in different shifts and days is not fixed, i.e. Equations (11)-(21). Taking this into consideration, we formulate the 0-1 programming model as follows:

Indices:

i = index of nurses;

j = index of weekdays;

k = index of shift types;

Sets:

I = index set of nurses;

I' = index set of nurses whose schedule is not fully occupied by AIP shifts;

J = set of indices of schedule days, $J = \{0(\text{Sun}), 1(\text{Mon}), 2(\text{Tue}), 3(\text{Wed}), 4(\text{Thu}), 5(\text{Fri}), 6(\text{Sat}), 7(\text{Sun})\}$;

J' = exclude 0(Sun) from set J ;

K = set of shift types, $K = \{1(\text{D}), 2(\text{E}), 3(\text{N}), 4(\text{X}), 5(\text{H}), 6(\text{AO}), 7(\text{O})\}$, where X is the equivalent of AL, CL, T, and ML.

User-defined inputs:

$l - r$ = lower and upper bounds of different types of nurses (Table I)

w_x = importance weight of soft constraint x ;

P = index set (i', j', k') indicating AIP day-offs k' of i' 'th nurse on j' 'th day;

N = set of number of total nurses, number of total NO, minimal numbers of different types of nurses in shift k on the j th day, i.e. $\{N_{ALL}(j, k), N_{NO}(j, k), N_{Nurse\ type}^{\min}(j, k)\}$. The values in the set can be found from the nurse coverage constraints currently adopted by the target ED (H_{4-1} – H_{4-4}) in Table IV. They are subject to change if necessary.

Decision variables:

$$X_{ijk} = \begin{cases} 1 & \text{if nurse } i \text{ is assigned to shift type } k \text{ for day } j \quad \forall i \in I, j \in J, k \in K \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Hard constraints:

H_1 : Day-offs approved in principle (AL, ML, CL, T).

$$X_{i'j'k'} = 1, \forall (i', j', k') \in P \quad (3)$$

H_{2-1} : Each nurse can only work one shift a day.

$$\sum_{k \in K} X_{ijk} = 1, \forall i \in I, \forall j \in J' \quad (4)$$

H_{2-2} : Each nurse must take one day-off (O) during the week.

$$\sum_{j \in J'} X_{ijk} = 1, \forall i \in I', k = 7 \quad (5)$$

H₂₋₃: Shift N must be preceded by D shift.

$$\sum_{k=2}^7 X_{ijk} + X_{i(j+1)k} < 2, \forall i \in I, j = 0, \dots, 6, k = 3 \quad (6)$$

H₃₋₁: Each nurse takes no more than one N shift during the week.

$$\sum_{j=J'} X_{ijk} \leq 1, \forall i \in I', k = 2 \quad (7)$$

H₃₋₂: Each nurse should not take more than three E shifts during the week.

$$\sum_{j \in J} X_{ijk} \leq 3, \forall i \in I, k = 2 \quad (8)$$

H₃₋₃: Three consecutive shift E must be avoided.

$$X_{ijk} + X_{i(j+1)k} + X_{i(j+2)k} < 3, \forall i \in I, j = 0, \dots, 5, k = 2 \quad (9)$$

H₃₋₄: H and AO should not be assigned to Sundays.

$$\sum_{k=5}^6 X_{ijk} = 0, \forall i \in I, j = 7 \quad (10)$$

H₄₋₁: Same number of nurses is assigned to D and E shifts. There are more nurses on Mondays and Tuesdays than other weekdays. Same number of nurses is assigned to N shifts over the week.

$$\sum_{i \in I} X_{ijk} = N_{ALL}(j, k), \forall j \in J', k = 1, 2, 3 \quad (11)$$

H₄₋₂: During N shifts, there must be a certain number of NO, a minimum number of MRN, FRN, and senior FRN.

$$\left\{ \begin{array}{l} \sum_{i=1}^l X_{ijk} = N_{NO}(j, k), \forall j \in J', k = 3 \\ \sum_{i=l+1}^o X_{ijk} \geq N_{MRN}^{\min}(j, k), \forall j \in J', k = 3 \\ \sum_{i=o+1}^r X_{ijk} \geq N_{FRN}^{\min}(j, k), \forall j \in J', k = 3 \\ \sum_{i=q+1}^r X_{ijk} \geq N_{FRN_S}^{\min}, \forall j \in J', k = 3 \end{array} \right. \quad (12-15)$$

H₄₋₃: During D shifts, there must be a certain number of NO, a minimum number of MRN and

senior FRN.

$$\left\{ \begin{array}{l} \sum_{i=1}^l X_{ijk} = N_{NO}(j,k), \forall j \in J', k=1 \\ \sum_{i=l+1}^o X_{ijk} \geq N_{MRN}^{\min}(j,k), \forall j \in J', k=1 \\ \sum_{i=q+1}^r X_{ijk} \geq N_{FRN_S}^{\min}, \forall j \in J', k=1 \end{array} \right. \quad (16-18)$$

H_{4.4}: During E shifts, there must be a certain number of NO, a minimum number of MRN and senior FRN.

$$\left\{ \begin{array}{l} \sum_{i=1}^l X_{ijk} = N_{NO}(j,k), \forall j \in J', k=2 \\ \sum_{i=l+1}^o X_{ijk} \geq N_{MRN}^{\min}(j,k), \forall j \in J', k=2 \\ \sum_{i=q+1}^r X_{ijk} \geq N_{FRN_S}^{\min}(j,k), \forall j \in J', k=2 \end{array} \right. \quad (19-21)$$

Soft constraints:

$$S_1 : \left\{ X_{ij2} + \sum_{k=5}^7 X_{i(j+1)k} \right\} - (s1_{ij}^+ - s1_{ij}^-) = 1, \forall i \in I; j = 0, \dots, 6, \quad (22)$$

where $s1_{ij}^+$ and $s1_{ij}^-$ denotes the positive and negative deviation from the result of Equation (22). If day-off (H, AO, O) is preceded by E shift, $s1_{ij}^+ = 1$. Only positive deviations are penalized.

$$S_2 : \left\{ X_{ij3} + \sum_{k=1}^5 X_{i(j+1)k} \right\} - (s2_{ij}^+ - s2_{ij}^-) = 1, \forall i \in I; j = 0, \dots, 6, \quad (23)$$

where $s2_{ij}^+$ and $s2_{ij}^-$ denotes the positive and negative deviation from the result of Equation (23). If shifts other than AO and O assigned after N shift, $s2_{ij}^+ = 1$. Only positive deviations are penalized.

$$S_3 : \left\{ X_{ij2} + X_{i(j+1)2} + X_{i(j+2)2} \right\} - (s3_{ij}^+ - s3_{ij}^-) = 2, \forall i \in I; j = 0, \dots, 5, \quad (24)$$

where $s3_{ij}^+$ and $s3_{ij}^-$ denotes the positive and negative deviation from the result of Equation (24). If three E shifts are assigned consecutively, $s3_{ij}^+ = 1$. Only positive deviations are penalized.

$$S_4 : \left\{ \sum_{k=4}^7 X_{ijk} + \sum_{k=1}^2 X_{i(j+1)k} + \sum_{k=4}^7 X_{i(j+2)k} \right\} - (s4_{ij}^+ - s4_{ij}^-) = 2, \forall i \in I; j = 0, \dots, 5, \quad (25)$$

where $s4_{ij}^+$ and $s4_{ij}^-$ denotes the positive and negative deviation from the result of Equation (25). If isolated day-on (off-on-off) appears, $s4_{ij}^+ = 1$. Only positive deviations are penalized.

Objective function:

$$\text{Min. } Z_{\text{benchmark}} = w_1 \sum_{i \in I} \sum_{j=0}^6 s1_{ij}^+ + w_2 \sum_{i \in I} \sum_{j=0}^6 s2_{ij}^+ + w_3 \sum_{i \in I} \sum_{j=0}^5 s3_{ij}^+ + w_4 \sum_{i \in I} \sum_{j=0}^5 s4_{ij}^+ \quad (26)$$

The result of each component in Equation (26) is actually the weighted sum of the corresponding number of soft constraint violations, so that the total value of Equation (26) is comparable to that of Equation (1), which is the objective function of the proposed approach. This 0-1 model is solved by ILOG CPLEX 10.0 on a PC with an Intel i5 2.26 GHz processor.

6. Experimental Results

In this section, we will compare the objective function value and time-taken of the schedules produced by the proposed approach with 0-1 programming. Also, comparisons will be made between the proposed approach and the existing approach. Next, since weight values of soft constraints are the reflection of group knowledge, it is necessary to prove that the fulfilment rate of soft constraints is independent on the weight values as long as the weight ranking is unchanged. In other words, the schedule quality are measured the same by different weight values. The proposed approach will be deemed robust if no significant quality variations are observed. Before making the above comparison, we first investigate which shift assignment sequence as discussed in *Section 4.1* can produce better initial schedules within shorter time. The two-sample t-test results as reported in Table VII show that if the *shift assignment heuristic* runs on the $N \rightarrow E \rightarrow D$ sequence, initial schedules with comparable ($p=0.38$) objective values can be generated but with less time ($p=0.00$) as compared to the $N \rightarrow D \rightarrow E$ sequence. Therefore, sequence $N \rightarrow E \rightarrow D$ is adopted in the first stage of our proposed approach. For all the statistical tests in this paper, a confidence level up to 95% is adopted.

[Table VII near here]

6.1 Benchmarking with 0-1 Programming

The quality of schedules generated by the proposed approach is measured by the average objective function value of ten *iterations* in each initial scenario. The difference between the average objective function values is tested by one-way ANOVA. The result in Table VIII indicates that average value of each scenario does not differ significantly from others ($p=0.46$). The fairly stable objective function values may be ascribable to the sequential search algorithm in the second stage of our approach, where the backlog of *unsolvable* lower-weighted soft constraints leads to a relatively smaller variation of objective function

value than higher-weighted ones would do. The total computing time and minimum objective function value of a 10-*iteration trial* is 10 to 14 minutes and 154 to 184, respectively. It should be noted that the total computing time is the sum of *shift assignment heuristic* and SLS. Due to the variety of initial scenarios, the time-taken of generating an initial solution can vary significantly. For example, the different initial scenarios in *trial 2* and *4* results in nearly 200-second difference. Obviously, more *iterations* in a *trial* will increase the probability of generating a better schedule. However, the prolonged computing time may not be affordable in the real ED environment. Taking this into consideration, we thus choose to run operate ten *iterations* a *trial* so that the total computing time is acceptable for most nurse officers.

[Table VIII near here]

Table VIII shows the comparisons the average objective function values between the proposed approach and 0-1 programming. We find that if some hard constraints (e.g. H_{3-1} – H_{3-4} , H_{4-1} – H_{4-4}) are relaxed, the optimal solutions can be identified within half hour by 0-1 programming. However, the search for the optimal solution without relaxation is very time-consuming in all scenarios. Considering that the optimal solution needs a long time to converge when the number of constraints and variables is large, we limit the computing time to 3 hours. Three major observations can be summarized from Table IX: 1) in 0-1 programming, the only optimal solution found within the time limit is scenario 2 and 9 only; 2) in scenario 1, 4, 5, 6, 7, better schedules are generated by 0-1 programming, but their optimality is not confirmed; 3) in scenario 3, 8, 10, better schedules are generated by the proposed approach but their optimality is not confirmed. Given the maximum computing time of 3 hours, the two-sample t-test shows that the schedules generated by the proposed approach possessing comparable quality with those by 0-1 programming in terms of average objective function value ($p=0.13$). In scenario 2, 9, optimal schedules are obtained by 0-1 programming with a lengthy computing time between 2.6 and 2.8 hours while the near-optimal schedules by the proposed approach only costs 0.19 to 0.22 hours. In the further tests which are not shown here, we find that in 8 out of 10 scenarios the optimal solutions can be found by 0-1 programming if there is no computing time limit. However, this merit of optimality assurance is overshadowed by the lengthy computing time which is not welcomed by nurse officers working under intense ED environment. Also, the nurse preferences are subject to change due to many unpredictable reasons such as nurse turnover sudden change of patient demands. Therefore, it is doubtful if optimality is a must for ED manager especially when the near-optimal schedules are good enough to cover most of the nurse preferences. Moreover, we have stressed from the beginning of this paper that automating the nurse scheduling process on Excel spreadsheet is motivated by the consideration of user-friendliness. Excel is a familiar working interface to all the nurses and the change of algorithms can be easily done by ED IT technicians without domain knowledge such as 0-1 programming. Last and most importantly,

implementing an Excel-based scheduling system does not require significant setup time and implementation cost as compared to a sophisticated algorithm built in commercial software. The concern of cost issue will be raised especially when the scheduling system is considered for implementation in multiple EDs under limited budget. Taking all these factors into account, the proposed approach is preferable to 0-1 programming in the environment of target ED.

[Table IX near here]

6.2 Comparison with the existing approach

The weekly schedule generated by the proposed approach and the one by the existing approach are compared in consideration of efficiency and schedule quality. As mentioned in Section 3.2, at least 2 hours are required to generate a feasible schedule while the realization of soft constraints is not always guaranteed, it means that only hard constraints are strictly fulfilled. On the contrary, Table IX reports that the average computing time of the proposed approach over 10 scenarios is only 0.19 hours or 11.4 minutes (range from 0.17 – 0.22 hours). As compared to the existing (manual) approach, the reduction of scheduling time is about $(2-0.19)/2 \times 100 = 90.5\%$. Moreover, the proposed approach can generate a feasible schedule by satisfying as many soft constraints as possible within a much shorter period of time. Thus, the schedule generated by our approach is obviously better than that of the existing approach in terms of both efficiency and quality (i.e. healthiness factor defined in Section 3.4). In this connection, the quality of healthcare service would be enhanced if the well-being of nurses can be secured.

6.3 Schedule Robustness

To measure the schedule robustness, we calculate the objective function value under different weight schemes. One-way ANOVA test is performed on the following hypothesis:

H₁: The *fulfilment rates* of soft constraints are the same under different weight schemes.

The hypothesis is tested on five sets of weight schemes on one initial scenario. The average *fulfilment rate* of each soft constraint across ten *iterations* is reported in Table X. For each *iteration*, the *fulfilment rate* of a soft constraint is determined by Equation (27). Weight values of soft constraints except S_2 (with the largest weight) are changed while keeping their weight ranking ($S_2 \rightarrow S_3 \rightarrow S_1, S_4$) in five different weight schemes. By doing so, we intended to show that the quality of schedule varies largely with respect to the importance ranking of soft constraints, rather than the exact value of weights.

[Table X near here]

$$\text{fulfillment rate of } S_x = \frac{\text{number nurses with } S_x \text{ violation}(s)}{\text{Total number of nurses}}, x = 1, 2, 3, 4 \quad (27)$$

In Table X, hypothesis H_1 is accepted by one-way ANOVA test that the average *fulfillment rates* of four soft constraints do not vary significantly between five weight schemes ($p > 0.05$). This indicates that as long as the objective function value is calculated on the same weight ranking, the quality of final schedules is the same regardless of different weight values. The percentages are reflective of the sequential nature of the proposed approach which results in more violations of lower-weighted soft constraints than higher-weighted ones. For example, the most desirable constraint S_2 is perfectly satisfied under five weight schemes, while the least desirable constraints S_1 and S_4 only have average rate of 67% and 72%, respectively.

In summary, our two-stage heuristic approach is able to produce quality schedules in a timely fashion. In comparison with 0-1 programming, our proposed approach has the advantage of user-friendliness and efficient generation of quality schedules with less computing time. The quality of schedules is also tested invariant under different weight values of soft constraints.

7. Conclusion and future work

This paper addressed a real NSP in a Hong Kong ED using a two-stage heuristic approach. As opposed to the manual scheduling, the proposed approach can generate quality schedules addressing both hard and soft constraints efficiently. This two-stage approach consists of *shift assignment heuristic* and SLS. The former generates initial schedules satisfying all hard constraints, and the latter makes refinement on the initial ones. This approach greatly reduces the computational complexity so that the scheduling process can be operated by VBA on Excel spreadsheet. Although the optimality of schedules is not guaranteed using the proposed approach as compared with 0-1 programming, good solutions were obtained within much less time. The proposed approach is also advantageous from several practical perspectives. First, it is able to reschedule at anytime in the middle of a scheduling period. This is very useful in an ED environment where frequent changes of shift/day-off are observed. Second, the characteristics of user-friendliness and cost-effectiveness are favoured by ED staff who are working under great pressure and tight budget. Given these reasons, the proposed approach is preferable to 0-1 programming if it will be considered for implementation in multiple EDs.

The future work will be dedicated to the following aspects. First, improvement can be made by considering other soft constraints which are not discussed in this paper, such as the balance of day and evening shifts across the week, and balanced assignment of weekend day-offs among nurses. Second, the algorithm of task assignment (Figure 1) in each shift, such as resuscitation, triage, etc. should be developed upon the existing algorithm. Both improvements will result in significant increase of computational burden such that the current algorithm may be no longer efficient in solution search. Therefore, more efforts should be

made to the development of more sophisticated and efficient heuristics.

Appendix I. Overall algorithm of sequential local search

I-A: Pseudo-code of shift assignment heuristic

```

While (a feasible initial solution has not yet found) {
  Randomly assign N shifts from Mon to Sun by  $H_{3-1}, H_{4-1}, H_{4-2}$ ;
  Randomly assign E shifts from Mon to Sun by  $H_{3-2}, H_{3-3}, H_{4-1}, H_{4-4}$ ;
  If (number of unassigned nurses from Mon to Sun > corresponding D shift coverage ) {
    Randomly assign D shifts for Mon-Sun by  $H_{2-3}, H_{4-1}, H_{4-3}$ ;
    If (at least one day is undetermined for all nurses) {
      Randomly assign O day-off by  $H_{2-2}$ ;
    }
    Else start over from While;
  }
  Else start over from While;
}

For (ith nurse of the total m) {
  If (his/her shifts/day-offs of Mon-Sun are not assigned) {
    For (jth unassigned shift/day-off of the total n) {
      If (his/her counter of AO > counter of H) {
        Assign an AO by  $H_{3-4}$ ;
      }
      If (his/her counter of AO = counter of H) {
        Assign either an AO or H by  $H_{3-4}$ ;
      }
      If (his/her counter of AO < counter of H) {
        Assign an H by  $H_{3-4}$ ;
      }
    }
    Until all unassigned shifts/day-offs are checked;
  }
  Until all nurses are checked;
}
}
Until a feasible initial solution is found;
}

```

I-B: Pseudo-code of sequential local search algorithm

i = rank of soft constraint by weight, $i \in (1, \dots, m)$, $m = 4$ in this case
 j = index of the violation area of i th soft constraint, $j \in (1, \dots, n_i)$
 $i = 1$, S = initial value of objective function

```

While  $i \leq m$ 
   $j = 1$ 
  While  $j \leq n_m$ 
    Calculate S for all possible swaps, i.e.  $S_1, S_2, S_3, \dots$ 
     $S_{\text{new}} = \text{Max} \{0, (S - S_1), (S - S_2), (S - S_3), \dots\}$ 
    If  $S_{\text{new}} > 0$ , Then
       $S = S_{\text{new}}$ 
    Else
       $j$ th violation area of  $i$ th soft constraint is temporarily unsolvable
  }
   $j++$ 

```

Appendix II. Three Basic Operators of sequential local search

II-A: Illustration of *vertical swap* operator imposed on S_3 violation. Any of the three E shifts in the *violation area* can be the *swap position*.

Nurses	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Nurse_1							
Nurse_2		D			E	E	E
Nurse_3							
Nurse_4							
Nurse_5					D		
Nurse_6							
Nurse_7	E	E	E				
Nurse_8							
Nurse_9							
Nurse_10							

II-B: Illustration of *double swap_1* operator imposed on S_2 violation. Any of the two shifts in the *violation area* can be the *swap position*.

Nurses	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Nurse_1							
Nurse_2		E					O
Nurse_3							
Nurse_4						N	E
Nurse_5		O					
Nurse_6							
Nurse_7			O		N		
Nurse_8							
Nurse_9							
Nurse_10			N		O		

II-C: Illustration of *double swap_2* operator imposed on S_4 violation. If the middle shift is E, the *swap position* can only be the second day-off because the swap between the first day-off and the middle shift can trigger S_4 violation. If the middle shift is D, both day-offs can be the *swap position*.

Nurses	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Nurse_1							
Nurse_2		H	E				
Nurse_3							
Nurse_4	AO	E	H				
Nurse_5							
Nurse_6							
Nurse_7		D	O				
Nurse_8							
Nurse_9							
Nurse_10		O	D	H			

References

- [1] Sarasohn-Kahn, J., 2012. It's the prices and the technology, stupid: why U.S. health costs are higher than anywhere in the world [online]. Available from: <http://healthpopuli.com/2012/05/07/its-the-prices-and-the-technology-stupid-why-u-s-health-costs-are-higher-than-anywhere-in-the-world/> [Accessed 6 August 2012].
- [2] Ernst, A.T., Jiang, H., Krishnamoorthy, M., and Sier, D., 2004. Staffing scheduling and rostering: A review of applications, methods, and models. *European Journal of Operational Research*, 153 (1), 3-27.
- [3] Chan, F.T.S., Wong, T.C., and Chan, L.Y., 2004. A genetic algorithm-based approach to machine assignment problem. *International Journal of Production Research*, 43 (12), 2451-2472.
- [4] Chan, F.T.S., Wong, T.C., and Chan, L.Y., 2006. Flexible job-shop scheduling problem under resource constraints. *International Journal of Production Research*, 44 (11), 2071-2089.
- [5] Chan, F.T.S., Wong, T.C., and Chan, L.Y., 2009. The application of genetic algorithms to lot streaming in job-shop scheduling problem. *International Journal of Production Research*, 47 (12), 3387-3412.
- [6] Lin, C.H., Hwang, S.L., and Wang, E.M.Y., 2007. A reappraisal on advanced planning and scheduling systems. *Industrial Management and Data Systems*, 107 (8), 1212-1226.
- [7] Ivert, L.K. and Jonsson, P., 2010. The potential benefits of advanced planning and scheduling systems in sales and operations planning. *Industrial Management and Data Systems*, 110 (5-6), 659-681.
- [8] Yan, S.Y. and Tu, Y.P., 2002. A network model for airline cabin crew scheduling. *European Journal of Operational Research*, 140 (3), 531-540.
- [9] Gopalakrishnan, B. and Johnson, E.L., 2005. Airline crew scheduling: State-of-the-art. *Annals of Operations Research*, 140 (1), 305-337.
- [10] Guo, Y.F., Mellouli, T., Suhl, L., and Thiel, M.P., 2006. A partially integrated airline crew scheduling approach with time-dependent crew capacities and multiple home bases. *European Journal of Operational Research*, 171 (3), 1169-1181.
- [11] Yen, J.W. and Birge, J.R., 2006. A stochastic programming approach to the airline crew scheduling problem. *Transportation Science*, 40 (1), 3-14.
- [12] Deng, G.F. and Lin, W.T., 2011. Ant colony optimization-based algorithm for airline crew scheduling problem. *Expert Systems with Applications*, 38 (5), 5787-5793.
- [13] Saddoune, M., Desaulniers, G., Elhallaoui, I., and Soumis, F., 2011. Integrated airline crew scheduling: A hi-dynamic constraint aggregation method using neighborhoods. *European Journal of Operational Research*, 212 (3), 445-454.
- [14] Valouxis, C. and Housos, E., 2002. Combined bus and driver scheduling. *Computers and Operational Research*, 29 (3), 243-259.

- [15] De Leone, R., Festa, P., and Marchitto, E., 2011a. Solving a bus driver scheduling problem with randomized multistart heuristics. *International Transactions in Operational Research*, 18 (6), 707-727.
- [16] De Leone, R., Festa, P., and Marchitto, E., 2011b. A bus driver scheduling problem: a new mathematical model and a GRASP approximate solution. *Journal of Heuristics*, 17 (4), 441-466.
- [17] Avramidis, A.N., Chan, W., Gendreau, M., L'Ecuyer, P., and Pisacane, O., 2010. Optimizing daily agent scheduling in a multiskill call center. *European Journal of Operational Research*, 200 (3), 822-832.
- [18] Robbins, T.R. and Harrison, T.P., 2010. A stochastic programming model for scheduling call centers with global service level agreement. *European Journal of Operational Research*, 207 (3), 1608-1619.
- [19] Xu, M., Wong, T. C., and Chin, K. S., 2013. Modeling daily patient arrivals at Emergency Department and quantifying the relative importance of contributing variables using artificial neural network. *Decision Support Systems*, 54 (3), 1488-1498.
- [20] Sitompul, D. and Randhawa, S.U., 1990. Nurse scheduling models: a state-of-the-art review. *Journal of the Society for Health Systems*, 2 (1), 62-72.
- [21] Marchionno, P.M., 1987. Modified cyclical scheduling: A practical approach. *Nursing Management*, 18 (10), 60-66.
- [22] Jaumard, B., Semet, F., and Vovor, T., 1998. A generalized linear programming model for nurse scheduling. *European Journal of Operational Research*, 107 (1), 1-18.
- [23] Dowsland, K.A. and Thompson, J.M., 2000. Solving a nurse scheduling problem with knapsacks, networks and tabu search. *Journal of the Operational Research Society*, 51 (7), 825-833.
- [24] Valouxis, C. and Housos, E., 2000. Hybrid optimization techniques for the workshift and rest assignment of nursing personnel. *Artificial Intelligence in Medicine*, 20 (2), 155-175.
- [25] Bard, J.F. and Purnomo, H.W., (2005b). Preference scheduling for nurses using column generation. *European Journal of Operational Research*, 164 (2), 510-534.
- [26] Belien, J. and Demeulemeester, E., 2007. A branch-and-price approach for integrating nurse and surgery scheduling. *European Journal of Operational Research*, 189 (3), 652-668.
- [27] Glass, C.A. and Knight, R.A., 2010. The nurse rostering problem: A critical appraisal of the problem structure. *European Journal of Operational Research*, 202 (2), 379-389.
- [28] Valouxis, C., Gogos, C., Goulas, G., Alefragis, P., and Housos, E., 2012. A systematic two phase approach for the nurse rostering problem. *European Journal of Operational Research*, 219 (2), 425-433.
- [29] Gascon, V., Villeneuve, S., Michelon, P., and Ferland, J.A., 2000. Scheduling the flying squad nurses of a hospital using a multi-objective programming model. *Annals of Operations Research*, 96 (1-4), 149-166.

- [30] Azaiez, M.N. and Al Sharif, S. S., 2005. A 0-1 goal programming model for nurse scheduling. *Computers and Operations Research*, 32 (3), 491-507.
- [31] Parr, D. and Thompson, J.M., 2007. Solving the multi-objective nurse scheduling problem with a weighted cost function. *Annals of Operations Research*, 155 (1), 279-288.
- [32] Burke, E.K., Li, J.P., and Qu, R., 2010. A hybrid model of integer programming and variable neighbourhood search for highly-constrained nurse rostering problems. *European Journal of Operational Research*, 203 (2), 484-493.
- [33] Moz, M. and Pato, M.V., 2004. Solving the problem of rerostering nurse schedules with hard constraints: New multicommodity flow models. *Annals of Operations Research*, 128 (1-4), 179-197.
- [34] Duenas, A., Tutuncu, G.Y., and Chilcott J.B., 2009. A genetic algorithm approach to the nurse scheduling problem with fuzzy preferences. *IMA Journal of Management Mathematics*, 20 (4), 369-383.
- [35] Burke, E.K., Curtois, T., Post, G., Qu, R., and Veltman, B., 2008. A hybrid heuristic ordering and variable neighbourhood search for the nurse rostering problem. *European Journal of Operational Research*, 188 (2), 330-341.
- [36] Kırış, S., Yüzügüllü, N., Ergünc, N., Çevikc, A.A., 2010. A knowledge-based scheduling system for Emergency Departments. *Knowledge-Based Systems*, 23 (8), 890-900.
- [37] Brucker, P., Burke, E.K., Curtois, T., Qu, R., and Vanden Berghe, G., 2010. A shift sequence based approach for nurse scheduling and a new benchmark dataset. *Journal of Heuristics*, 16 (4), 559-573.
- [38] Burke, E.K., De Causmaecker, P., Petrovic, S., and Vanden Berghe, G., 2006. Metaheuristics for handling time interval coverage constraints in nurse scheduling. *Applied Artificial Intelligence*, 20 (3), 743-766.
- [39] Díaz, J.A. and Fernández, E., 2001. A tabu search heuristic for the generalized assignment problem. *European Journal of Operational Research*, 132 (1), 22-38.
- [40] Burke, E.K., Kendall, G., and Soubeiga, E., 2003. A tabu search hyperheuristic for timetabling and rostering. *Journal of Heuristics*, 9 (6), 451-470.
- [41] Bellanti, F., Carello, G., Croce, F.D., and Tadei, R., 2004. A greedy-based neighborhood search approach to a nurse rostering problem. *European Journal of Operational Research*, 153 (1), 28-40.
- [42] Aickelin, U., Burke, E.K., and Li, J., 2007. An estimation of distribution algorithm with intelligent local search for rule-based nurse rostering. *Journal of the Operational Research Society*, 58 (12), 1574-1585.
- [43] Bester, M.J., Nieuwoudt, I., and Van Vuuren Jan, H., 2007. Finding good nurse duty schedules: A case study. *Journal of Scheduling*, 10 (6), 387-405.
- [44] Maenhout, B. and Vanhoucke, M., 2007. An electromagnetic meta-heuristic for the nurse scheduling problem. *Journal of Heuristics*, 13 (4), 359-385.

- [45] Burke, E.K., Curtois, T., van Draat, L.F., van Ommeren, J.K., and Post, G., 2011. Progress control in iterated local search for nurse rostering. *Journal of the Operational Research Society*, 62 (2), 360-367.
- [46] Bilgin, B., De Causmaecker, P., Rossie, B., Vanden Berghe, G., 2012. Local search neighbourhoods for dealing with a novel nurse rostering model. *Annals of Operations Research*, 194 (1), 33-57.
- [47] Lu, Z.P. and Hao, J.K., 2012. Adaptive neighborhood search for nurse rostering. *European Journal of Operational Research*, 218 (3), 865-876.
- [48] Dowsland, K.A., 1998. Nurse scheduling with tabu search and strategic oscillation. *European Journal of Operational Research*, 106 (2-3), 393-407.
- [49] Bard, J.F. and Purnomo, H.W., 2005a. Hospital-wide reactive scheduling of nurses with preference considerations. *IIE Transactions*, 37 (7), 589-608.
- [50] Bertels, S. and Fahle, T., 2006. A hybrid setup for a hybrid scenario: combining heuristics for the home health care problem. *Computers and Operations Research*, 33 (10), 2866-2890.
- [51] Mladenovic, N. and Hansen, P., 1997. Variable neighborhood search. *Computers and Operations Research*, 24 (11), 1097-1100.
- [52] Hansen, P. and Mladenovic, N., 2001. Variable neighborhood search: Principles and applications. *European Journal of Operational Research*, 130 (3), 449-467.
- [53] Kundu, S., Mahato, M., Mahanty, B., Acharyya, S., 2008. Comparative performance of simulated annealing and genetic algorithm in solving nurse scheduling problem. In proceedings of the IMECS International Multiconference of Engineers and computer scientists, Hong Kong, 2008, 1680-1686.
- [54] Gutjahr, W.J. and Rauner, M.S., 2007. An ACO algorithm for a dynamic regional nurse-scheduling problem in Austria. *Computers and Operations Research*, 34 (3), 642-666.
- [55] Cheang, B., Li, H., Lim, A., and Rodrigues, B., 2003. Nurse rostering problems: a bibliographic survey. *European Journal of Operational Research*, 151 (3), 447-460.
- [56] Burke, E.K., De Causmaecker, P., Vanden Berghe, G., and Van Landeghem, H., 2004. The state of the art of nurse rostering. *Journal of Scheduling*, 7 (6), 441-499.
- [57] Colomi, A., Dorigo, M., and Maniezzo, V., 1991. Genetic algorithms and highly constrained problems: The time-table case. *Lecture Notes in Computer Science*, 496, 55-59.
- [58] Aickelin, U. and Dowsland, K., 2000. Exploiting problem structure in a genetic algorithms approach to a nurse rostering problem. *Journal of Scheduling*, 3 (3), 139-153.
- [59] Kawanaka, H., Yamamoto, K., Yoshikawa, T., Shinogi, T., and Tsuruoka, S., 2001. Genetic algorithm with the constraints for nurse scheduling problem. In Proceedings of the Congress on Evolutionary Computation, Seoul, South Korea, 2001, 1123-1130.

- [60] Aickelin, U. and Dowsland, K.A., 2004. An indirect genetic algorithm for a nurse scheduling problem. *Computers and Operations Research*, 31 (5), 761-778.
- [61] Aickelin, U. and White, P., 2004. Building better nurse scheduling algorithms. *Annals of Operations Research*, 128, 159-177.
- [62] Moz, M. and Pato, M.V., 2007. A genetic algorithm approach to a nurse rostering problem. *Computers and Operations Research*, 34 (3), 667-691.
- [63] Yeh, J.Y. and Lin, W.S., 2007. Using simulation technique and genetic algorithm to improve the quality care of a hospital emergency department. *Expert Systems with Applications*, 32 (4), 1073-1083.
- [64] Puente, J., Gomez, A., Fernandez, I., and Priore, P., 2009. Medical doctor rostering problem in a hospital emergency department by means of genetic algorithms. *Computers and Industrial Engineering*, 56 (4), 1232-1242.
- [65] Tsai, C.C. and Li, S.H. A., 2009. A two-stage modeling with genetic algorithms for the nurse scheduling problem. *Expert Systems with Applications*, 36 (5), 9506-9512.
- [66] Maenhout, B. and Vanhoucke, M., 2008. Comparison and hybridization of crossover operators for the nurse scheduling problem. *Annals of Operations Research*, 159 (1), 333-353.
- [67] Van de Ven, A.H. and Delbecq, A.L., 1974. The effectiveness of nominal, delphi, and interacting group decision making process. *Academy of Management Journal*, 17 (4), 605-621.
- [68] Delbecq, A.L. and Van de Ven, A.H., 1986. *Group techniques for program planning: a guide to nominal group and Delphi processes*, Wisconsin: Green Briar.
- [69] Oldenkamp, J.H., 1996. Quality factors of nursing schedules. In: Wever, H., *Quality in Fives, Labyrinth Publicatoin*, The Neitherlands, 53-70.

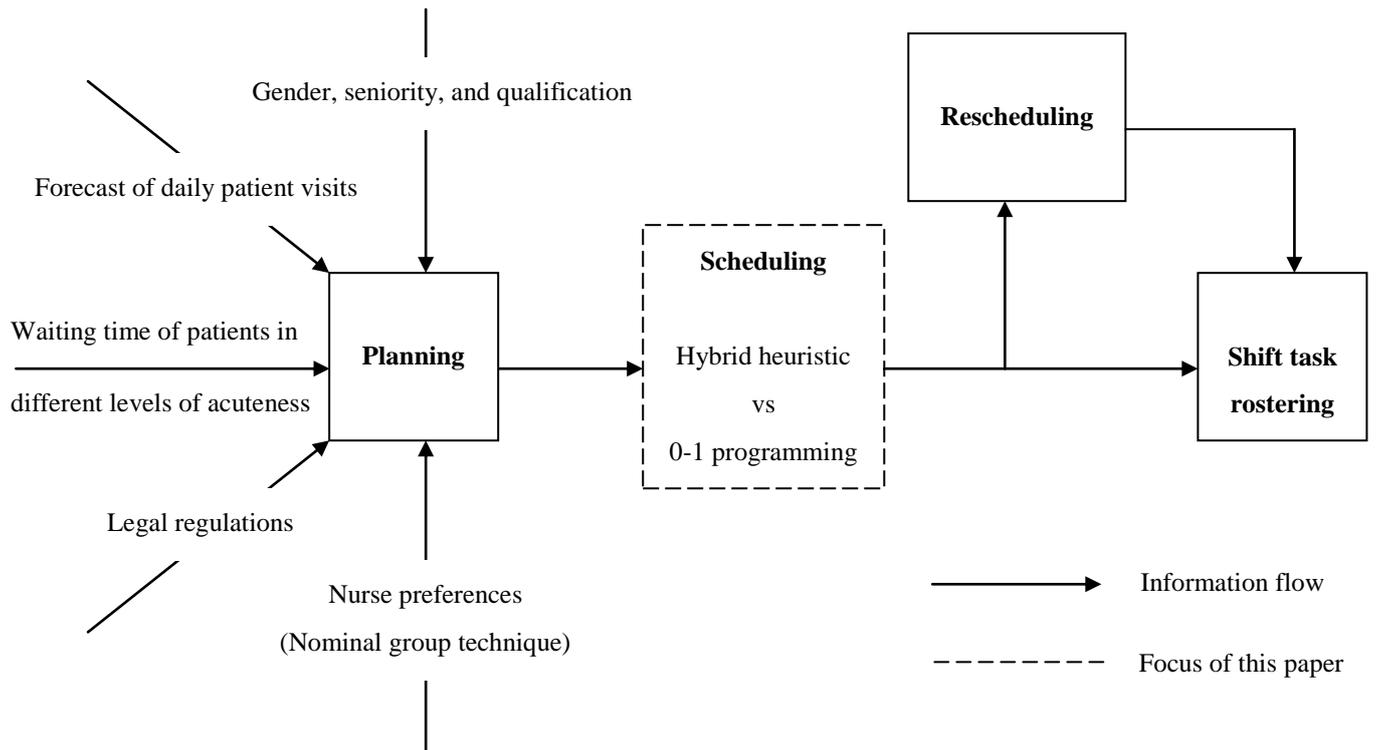


Figure 1. Nurse scheduling procedures for the target ED

$H_{2-2}, H_{2-3}, H_{3-1}, H_{3-2}, H_{3-3}, H_{3-4}$, and all soft constraints

		Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun
$H_{4-1}, H_{4-2}, H_{4-3}, H_{4-4}$	Nurses								
	NO_1	D	E	D	H	H	D	O	E
	NO_2	O	E	E	E	D	D	D	O
	NO_3	E	O	D	N	E	E	D	D
	NO_4	O	O	D	D	N	E	E	D
	NO_5	D	AL	AL	T	H	H	H	O
	NO_6	N	AO	O	D	H	H	D	N
	NO_7	O	D	O	E	D	D	N	E
	⋮								
	FRN_S_r-1	E	E	E	E	H	D	N	O
	FRN_S_r	O	D	D	E	D	E	O	E

Figure 2. Partial view of an initial schedule

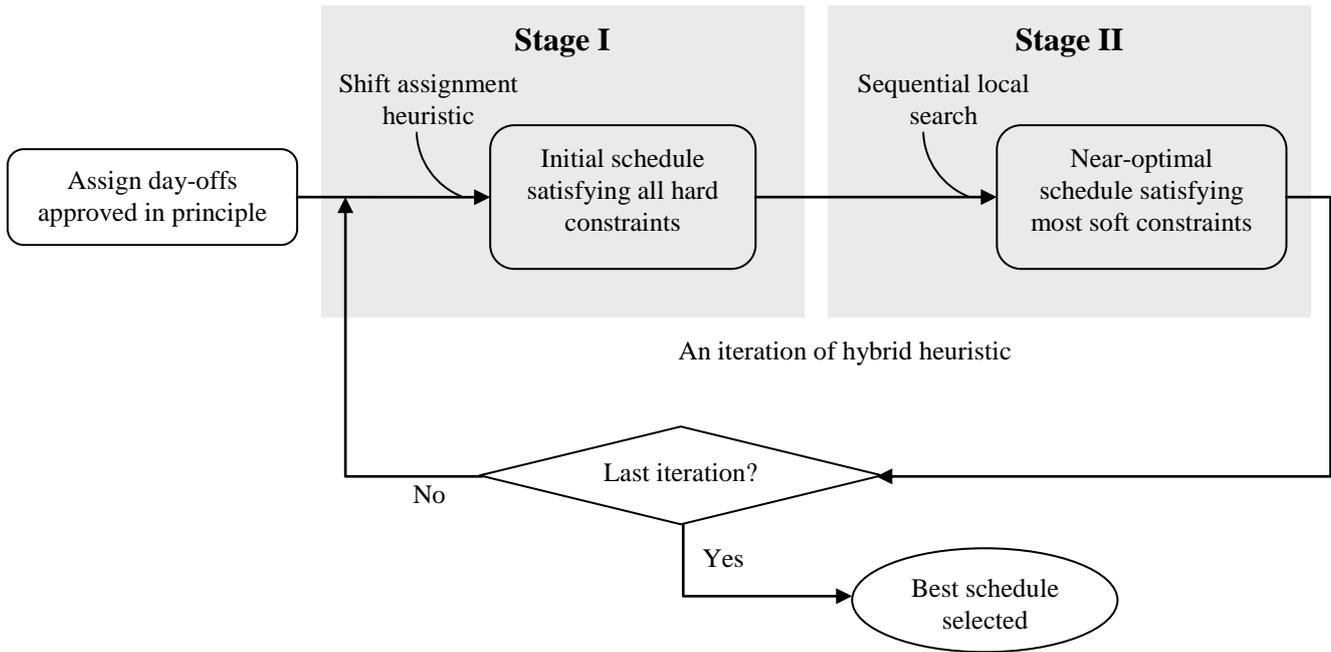


Figure 3. Conceptual diagram of the proposed approach

Table I. ED Nurse types in the target ED

Nurse type	Number range
NO: nurse officer	$1 - l$
MRN_J: male registered nurse at novice level	$l+1 - m$
MRN_M: male registered nurse at intermediate level	$m+1 - n$
MRN_S: male registered nurse at senior level	$n+1 - o$
FRN_J: female registered nurse at novice level	$o+1 - p$
FRN_M: female registered nurse at intermediate level	$p+1 - q$
FRN_S: female registered nurse at senior level	$q+1 - r$
Total:	r

Table II. Shifts and day-offs in the target ED

Shift and day-off	Description
D: Day shift	7am-3pm
E: Evening shift	3pm-11pm
N: Night shift	11pm-7am
AL: Annual leave	Vary between nurses
CL: Compassionate leave	Two consecutive days
ML: Maternity leave	Vary between nurses
T: Training class	Vary between nurses
O: Day-off	One day per week
H: Statutory and public holiday	Unused H can be reserved
AO: Accumulated day-off	Half day per week; unused AO can be reserved

Table III. Weight of nurse preferences obtained by nominal group technique

Nurse preference	N1	N2	N3	N4	N5	Total
1. Each nurse takes no more than one N shift during the week.	4	3		2		9
2. Each nurse should not take more than three E shifts during the week.	2	1	4	2		9
3. Three consecutive E shifts should be avoided.	2		3	2	3	10
4. AO and H should not be assigned to Sundays.	2	2		2	3	9
5. E shift precedes a day-off (H, O, AO) should be avoided.		1		1		2
6. Shifts other than O and AO assigned after N shift should be avoided.		1	1	1	2	5
7. Three consecutive E shifts should be avoided.		1	2		1	4
8. Isolated days on (off-on-off) should be avoided.		1			1	2
Total:	10	10	10	10	10	50

Table IV. The hard constraints for nurse scheduling

Hard constraints	
Pre-assignment	H ₁ : Day-offs approved in principle (AL, ML, CL, T) should be assigned before nurse scheduling.
Legal	H ₂₋₁ : Each nurse can only work one shift a day. H ₂₋₂ : Each nurse must take one day-off (O) during the week. H ₂₋₃ : N shift must be preceded by D shift.
Nurse preference	H ₃₋₁ : Each nurse takes no more than one N shift during the week. H ₃₋₂ : Each nurse should not take more than three E shifts during the week. H ₃₋₃ : Three consecutive shift E must be avoided. H ₃₋₄ : AO and H should not be assigned to Sundays.
Shift coverage	H ₄₋₁ : The daily coverage is 16 nurses for both D and E shift on all Mondays, and the corresponding number of Tuesdays and the rest weekdays are 15 and 14 respectively. 6 nurses for N shift in all weekdays and weekends. H ₄₋₂ : During N shifts, there must be one NO, at least one MRN, at least three FRN, at least one senior FRN. H ₄₋₃ : During D shifts, there must be three NO, at least one MRN, at least two senior FRN. H ₄₋₄ : During E shifts, there must be two NO, at least one MRN, at least two senior FRN.

Table V. The soft constraints for nurse scheduling

Soft constraints	Weight
S ₁ : E shift precedes a day-off (H, O, AO) should be avoided.	4
S ₂ : Shifts other than O and AO assigned after N shift should be avoided.	10
S ₃ : Three consecutive E shifts should be avoided.	8
S ₄ : Isolated days on (off-on-off) should be avoided.	4

Table VI. Soft constraints and the associated SLS operators

Soft constraint	Violation type (Num. of swap positions)	SLS operator
S ₁	E→day-off ^a (2)	<i>Double swap_1, Vertical swap</i>
S ₂	N→E (2) N→D (2)	<i>Double swap_1, Vertical swap</i>
S ₃	E→E→E (3)	<i>Double swap_1, Vertical swap</i>
S ₄	day-off→E→day-off (1) day-off→D→day-off (2)	<i>Double swap_2</i>

^aO, AO, or H

Table VII. A side-by-side comparison of two shift assignment sequences based on ten iterations for each initial scenario

Initial scenario	Average OFV ^a		Average time (second)	
	N→E→D	N→D→E	N→E→D	N→D→E
1	335	344	39	58
2	356	350	51	81
3	324	332	37	59
4	308	320	27	28
5	330	323	47	79
6	313	329	33	48
7	350	354	43	70
8	341	327	45	51
9	339	341	30	49
10	347	342	31	61
Two-sample t-test	p=0.38		p=0.00	

^aObjective function value

Table VIII. Experimental results of the proposed approach

Initial scenario	OFV ^a in 10 iterations										Average OFV	Minimum OFV	Total time (second)
	1	182	226	194	188	170	204	208	204	192	182	195	170
2	194	188	196	182	180	216	186	188	190	186	191	180	803
3	166	184	190	208	196	180	170	180	188	178	184	158	702
4	192	190	194	172	190	170	186	212	216	184	191	178	605
5	184	190	166	178	154	204	198	188	208	200	187	154	640
6	168	202	184	166	216	190	202	194	226	176	192	166	645
7	204	196	212	190	194	184	226	192	192	214	200	184	752
8	172	192	208	192	184	172	200	206	182	204	191	172	725
9	188	194	218	192	172	200	206	178	204	184	194	180	667
10	194	206	188	172	208	172	204	184	176	164	187	156	648
One-way ANOVA df (9, 90)											F=0.98	p=0.46	

^aObjective function value

Table IX. Comparison between the proposed approach and 0-1 programming

Scenario	0-1 programming		Hybrid heuristic	
	Computing time (hour)	OFV ^a	Computing time (hour)	Minimal OFV
1	3.0	152	0.18	170
2	2.6	162*	0.22	180
3	3.0	160	0.20	158
4	3.0	168	0.17	178
5	3.0	152	0.18	154
6	3.0	154	0.18	166
7	3.0	176	0.21	184
8	3.0	174	0.20	172
9	2.8	156*	0.19	180
10	3.0	176	0.18	156
Two-sample t-test	p=0.13			

*Optimality confirmed; ^aObjective function value

Table X. Average fulfilment rate of ten iterations under different weight schemes of soft constraints

Weight schemes (w_1, w_2, w_3, w_4)	Average fulfilment rate %			
	S ₁ (Std)	S ₂ (Std)	S ₃ (Std)	S ₄ (Std)
(4, 10, 8, 4)	68 (10.6)	100 (0)	85 (6.2)	72 (7.8)
(4, 10, 9, 4)	65 (8.1)	100 (0)	84 (5.6)	75 (8.9)
(4, 10, 7, 4)	64 (9.0)	100 (0)	86 (5.9)	69 (7.5)
(7, 10, 8, 7)	67 (8.1)	100 (0)	81 (6.8)	73 (9.1)
(1, 10, 8, 1)	71 (8.0)	100 (0)	83 (7.1)	71 (8.4)
One-way ANOVA df (4, 45)	F=1.09 p=0.37	F=1.00 p=0.42	F=1.50 p=0.22	F=1.45 p=0.23