

A NEW TWO-STAGE VARIABLE NEIGHBORHOOD SEARCH ALGORITHM FOR THE NURSE ROSTERING PROBLEM

MOHAMMED ABDELGHANY^{1,2,*}, ZAKARIA YAHIA² AND AMR B. ELTAWIL¹

Abstract. The nurse rostering problem refers to the assignment of nurses to daily shifts according to the required demand of each shift and day, with consideration for the operational requirements and nurses' preferences. This problem is known to be an NP-hard problem, difficult to be solved using the known exact solution methods especially for large size instances. Mostly, this problem is modeled with soft and hard constraint, and the objective is set to minimize the violations for the soft constraints. In this paper, a new two-stage variable neighborhood search algorithm is proposed for solving the nurse rostering problem. The first stage aims at minimizing the violations of the soft constraints with the higher penalty weights in the objective function. While the second stage considers minimizing the total solution penalty taking into account all the soft constraint. The proposed algorithm was tested on the 24 benchmark instances of Curtois and Qu (Technical Report, ASAP Research Group, School of Computer Science, University of Nottingham (2014)). The test results revealed that the proposed algorithm is able to compete with the results of a recent heuristic approach from literature for most of the tested instances.

Mathematics Subject Classification. 90C59, 90B35.

Received November 21, 2019. Accepted February 20, 2021.

1. INTRODUCTION

Humans are the key resource in the healthcare services sector. Although the continuous increase in demand for healthcare services, the available staff is still limited. That is why the staff scheduling problem is receiving the interest of many researchers. Nurses are one of these vital resources that have a great role in the quality of healthcare services. Furthermore, hospitals around the world are facing the prevailing nurse shortage problem [19]. Therefore, proper scheduling of such human resource is a must to guarantee the quality of the provided healthcare service.

The staff scheduling process usually passes through three main stages [15]. The first stage is the long-term stage, which focuses on the staffing levels and the employment strategies of the institution. The second is the mid-term planning stage that deals with the assignments of the staff to the daily shifts (staff rostering). The

Keywords. Nurse rostering, nurse scheduling, staff scheduling, variable neighborhood search, heuristics, timetabling.

¹ Industrial and Manufacturing Engineering, Egypt-Japan University of Science and Technology, New Borg El-Arab City, Alexandria 21934, Egypt.

² Mechanical Engineering Department, Fayoum University, Fayoum, Egypt.

*Corresponding author: mohammed.abdelgalil@ejust.edu.eg

last stage is the short-term planning stage, where the staff re-rostering problem is considered to cope with any changes in the original roster. The focus of this work will be on the second stage – the staff rostering problem.

The nurse rostering problem (NRP) can be defined as the assignment of nurses to the daily shifts according to the required demand for each shift [11]. There are many considerations that have to be taken into account while preparing a roster such as the hospital regulations, nurses' contracts, management preferences as well as nurses' preferences. The NRP is known to be an NP-hard problem [10]. The constraints of the NRPs are usually categorized into hard and soft constraints. The hard constraints are those constraints that must be satisfied for a solution/roster to be feasible. The soft constraints are those ones that may be violated but they are penalized in the objective function. Therefore, the objective function in the NRPs is basically to minimize the violations for these soft constraints as much as possible. More details regarding the hard and soft constraints of the NRPs will be provided in Section 3.

In this paper, we propose a new two-stage variable neighborhood search approach that divides the soft constraints into two main groups and tries to satisfy each group sequentially, starting by the constraints with the higher weights in the objective function. In this approach, we propose two new neighborhood search structures that focus on satisfying the soft constraints with the highest weights as much as possible.

The rest of this paper is organized as follows. A review of the recent literature discussing the nurse rostering problem is provided in Section 2. After that, a description of the nurse rostering problem is given in Section 3. Then the propose two-stage neighborhood search approach for solving the nurse rostering problem is illustrated in Section 4. In Section 5, computational results and evaluation for the proposed algorithm are presented. Finally, some conclusions and potential future research work are discussed in Section 6.

2. LITERATURE REVIEW

There are many published studies during the last decades discussing the nurse rostering problem. A variety of exact and approximate solution methods and algorithms were applied to solve this problem. A mixed integer programming model for a NRP in a Kuwaiti healthcare unit was developed by M'Hallah and Alkhabbaz [14] with the objective of minimizing the number of outsources nurses.

Hadwan *et al.* [11] used a harmony search algorithm to solve a real NRP of a Malaysian hospital. Martin *et al.* [13] proposed a cooperative search framework that combines the strengths of multiple metaheuristics (tabu search, simulated annealing and variable neighborhood search) for solving the NRP. A particle swarm optimization method was employed by Gao and Lin [9] to solve a NRP with the objective of maximizing the nurses work happiness. Another particle swarm optimization algorithm with some refinement procedures was used in [20] to maximize the fairness of allocation of different shift types and days between nurses over the planning horizon. A two-stage heuristic approach for nurse scheduling in an emergency department was proposed in [18]. The first stage aims to generate an initial schedule that satisfies all hard constraints. After that, a sequential local search algorithm is used to improve the initial schedule taking into account the soft constraints.

Burke *et al.* used a hybrid integer programming and variable neighborhood search approach to solve the NRP [6]. An integer programming (IP) model was used to solve the subproblems where only the less computing complexity set of constraints are considered. Then variable neighborhood search was used to improve the solution of the IP model by satisfying the remaining constraints that were not considered in the IP model. Another hybrid approach using integer programming and local search was proposed in [17]. In that work, the problem was solved on two stages. In the first stage; the nurses are assigned to the working days, while in the second stage; the nurses are assigned to the different shift types.

Indeed, there are many other studies where heuristic methods were adopted such as; the adaptive neighborhood search algorithm [22], an improved harmony search algorithm [2], a chemical reaction optimization heuristic [21], a hybrid artificial bee colony and hill climbing optimizer approach [3] and a two phase adaptive variable neighborhood search algorithm [16]. Indeed, due to the computational complexity of the NRP, heuristic-based methods are the most used to tackle this problem [1].

TABLE 1. A comparison between three benchmark instances in terms of hard and soft constraints.

Constraint description	INRC.I [12]	INRC.II [7]	Curtois [8]
Shifts' cover requirements	Hard	Soft	Soft
Single shift assignment per day	Hard	Hard	Hard
Unwanted shift patterns	Soft	Hard	Hard
Max/Min Number of assignments	Soft	Soft	Hard
Number of consecutive working days	Soft	Soft	Hard
Number of consecutive days-off	Soft	Soft	Hard
Number of working weekends	Soft	Soft	Hard
Days on/off requests	Soft	Soft	Hard
Shifts on/off requests	Soft	Soft	Soft
Alternative skills	Soft	Hard	N/A

Briefly, the domination of the heuristic-based methods is obvious in literature. This domination is mostly due to the computational complexity of the NRP. However, a major weakness of the heuristic-based methods is that they cannot guarantee the solution optimality nor reducing the search space. Therefore, this open the scope for proposing new solution methods that may succeed in achieving better results.

The available literature in this topic is either providing a real case study of the problem in certain hospitals such as the work presented in [11], or evaluating new solution methods for some NRP benchmark instances. There are multiple sets of instances available in literature such as those instances provided in the First [12] and the Second [7] International Nurse Rostering Competitions (INRC.I and INRC.II) and the recent 24 instances introduced in [8]. Although most of these benchmark instances share almost the same problem constraints, but the major difference between them is which constraint is considered a hard constraint and which is a soft one. Table 1 summarizes the main characteristics of the three NRP benchmark instances regarding the soft and hard constraints.

3. PROBLEM DESCRIPTION

In this paper, the benchmark instances introduced in [8] have been chosen for evaluating the proposed solution method. In this set of NRP instances, there is a large number of constraints to be satisfied. These constraints are categorized into two groups of hard constraints and soft constraints. The fewer soft constraints in these instances make them the most convenient to evaluate our proposed approach. Constraints with the highest penalty weights can be clearly identified. As well as, neighborhood search structures-based-specific soft constraints can be directly implemented.

Hard constraints

- Each nurse cannot be assigned more than one shift per day.
- *Shift rotation*; shift types that cannot follow a certain shift type on the previous day.
- *MaxShifts*; the maximum number of shifts of each shift type that can be assigned to a nurse during the planning horizon.
- *MaxTotalMinutes*; the maximum amount of time that can be assigned to a nurse during the planning horizon.
- *MinTotalMinutes*; the minimum amount of time that must be assigned to a nurse during the planning horizon.
- *MaxConsecShifts*; the maximum number of consecutive working shifts that can be assigned to a nurse before having a day-off.
- *MinConsecShifts*; the minimum number of consecutive working shifts that must be assigned to a nurse before having a day-off.

TABLE 2. List of parameters and notations.

N	Set of nurses (the total number of nurses)
n	Nurse index $\in N$, $n = \{1, \dots, N\}$
D	Set of days (the total number of days in the planning horizon)
d	Day index $\in D$, $d = \{1, \dots, D\}$
S	Set of shift types, including day_{off}
s	Shift index $\in S$, $s = \{1, \dots, S\}$
h	Number of violations for hard constraints
X	A complete solution/nurses roster
$X_{n,d}$	The assigned shift type for a nurse n on day d
O_t	Total soft constraints violations penalty
O_c	Covering constraint violations penalty
SH	Set of shifts with cover shortage sorted descendingly according to their penalty values
$sh_{s,d}$	Cover shortage penalty value for a shift type s on day d
off_d	Set of nurses that are off (have no assignment) on day d
Wd_n	Set of already assigned working days for nurse n
AT_n	Available time for nurse n
$AS_{s,n}$	Available shifts of type s that can be assigned for nurse n

- $MinConsecDays-off$; the minimum number of consecutive days-off that must be assigned to a nurse before assigning a working shift.
- $MaxWeekends$; the maximum number of working weekends during the planning horizon.
- $Days-off$; shifts cannot be assigned to certain nurses on certain days.

Soft constraints

- *Shift on requests*; if a nurse is not assigned a certain shift on a specific day that he/she requested to work, a solution penalty is applied.
- *Shift off requests*; if a nurse is assigned a certain shift on a specific day that he/she requested not to work on, a solution penalty is applied.
- *Cover*; if the required number of nurses on a specific shift of a specific day is not assigned, a solution penalty is applied. If the number of the assigned nurses is less than the required number, the penalty is calculated as follows; $(required\ number\ of\ nurses - assigned\ number\ of\ nurses) * weight\ for\ shortage$. While if the number of the assigned nurses is greater than the required number, the penalty is calculated as follows; $(assigned\ number\ of\ nurses - required\ number\ of\ nurses) * (weight\ for\ surplus)$.

The objective of the NRP is to minimize the total penalty resulting from the soft constraints violations. For a detailed description of the problem, its constraints and an integer programming formulation for it, the reader is referred to [8]. Before proceeding to the proposed algorithm description, a list of different parameters and notations to be used through the rest of this paper is presented in Table 2.

4. THE PROPOSED TWO-STAGE VARIABLE NEIGHBORHOOD SEARCH APPROACH

In this section, we describe a two-stage variable neighborhood search approach to solve the described nurse rostering problem. A general overview of the proposed approach is demonstrated in Figure 1.

Firstly, some algorithm parameters are initialized and the termination conditions for each stage of the algorithm are defined. Also, some input data pre-processing is performed. All the alphabetic representations of nurses' indices or shift types are converted into numerical indices for easier coding and data processing purposes.

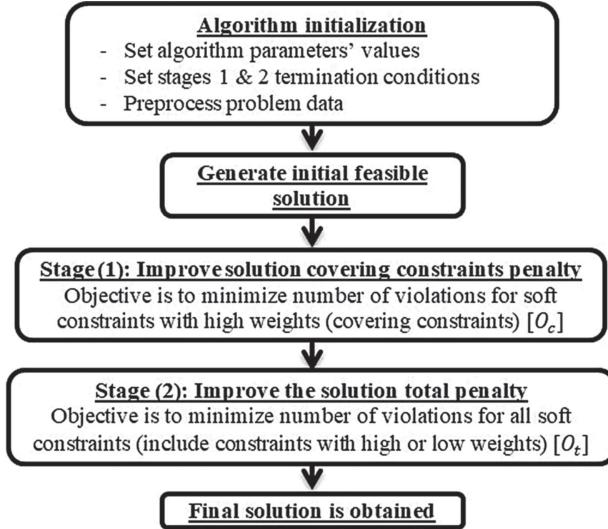


FIGURE 1. A general overview of the proposed approach.

4.1. Initial solution construction

After input data pre-processing, an initial feasible solution is generated using the greedy heuristic algorithm described by the pseudo code in Figure 2. For each nurse a random shift assignment is done on a day by day basis from the first day till the last day of the planning horizon. After each assignment, the feasibility of the partial solution of the nurse is checked against the applicable hard constraints. It is worth mentioning that the evaluation of some of the hard constraints is postponed either to the last day of the planning horizon *e.g.* minimum total minutes or till reaching a day whose index is greater than the constraint limit *e.g.* maximum consecutive shifts. For example, it is worthless evaluating the maximum consecutive shifts constraint in day 3 while the maximum allowed number of consecutive working days is 5. For each day, if no feasible assignment is found, the day is left off and any affected preceding few assignments that may cause some constraint violations are set to be off. The evaluation of the *MinTotalMinutes* constraint is left till the last day. After that, for each nurse assignments with *MinTotalMinutes* constraint violation, another step (4.1) in the shown pseudo code is used to eliminate this violation. In this step, a set of horizontal swaps – explained in Section 4.2.2 – are used to perturb the nurse schedule, followed by random shift assignments on off days to satisfy the *MinTotalMinutes* constraint. Finally, the complete feasible assignments that are generated for each nurse are merged together to form the initial full solution X . After that, the two-stage variable neighborhood search is applied as will be detailed in the following sections.

4.2. Neighborhood search structures

There are several neighborhood structures to be used in both stages of the proposed algorithm. These structures can be categorized into two main groups as follows:

4.2.1. High-Weight-Constraints-Focused structures (HWCF)

This HWCF group consists of two new neighborhood search structures that consider only the movements which may cause great improvements in the objective function. This is achieved by focusing only on soft constraints with the higher weights in the objective function. According to the instances' data of the problem under study, the highest weight is always associated with the shortages in satisfying the required number of nurse in a certain shift (covering constraint). Therefore, the focus of the proposed two structures is to minimize

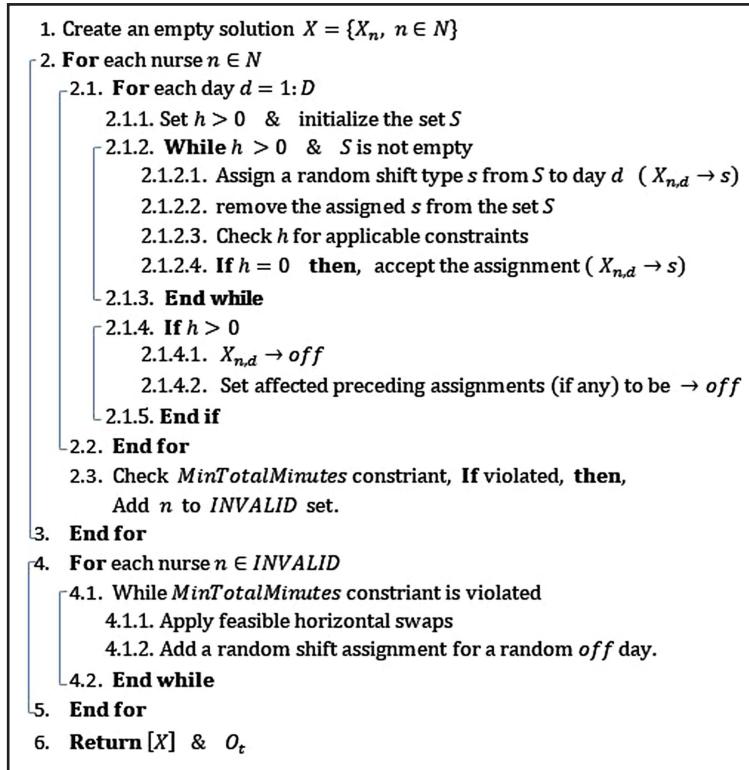


FIGURE 2. Pseudo code for initial feasible solution construction algorithm.

these shortages in covering as much as possible. A key feature of the proposed two structures is ordering of shifts with cover shortage descendingly according to their penalty values in set SH. So, shifts with higher cover shortage penalty will have a priority to be satisfied before shifts with lower penalty values. Therefore, rapid improvement can be achieved by satisfying large shortages in shifts first.

HWCF.1: The aim of this structure is to utilize the available amount of time that can be assigned to each nurse by making more feasible assignments in order to minimize the cover shortages. There are two main conditions that must be satisfied for that structure to be applicable. First, the available time for the nurse must be longer than the duration of the intended shift to be assigned, where the available time is the difference between the allowable $MaxTotalMinutes$ for the nurse and the actual assigned time. Secondly, the available number of shifts of the intended shift type that can be assigned to the nurse must be greater than zero, where the available number of shifts of a certain type for a nurse is the difference between the allowed $MaxShifts$ and the number of times this shift type has been already assigned to the nurse. The pseudo code in Figure 3 summarizes the neighborhood search process of HWCF.1 structure.

An illustrative example for HWCF.1 structure is shown in Figure 4. The tables on top represent the cover violations penalty value for each shift type (E , D and N) at each day. As per the studied problem, the weight for a cover shortage is 100 and for a surplus is 1. Therefore, the penalty values greater than or equal to 100 represent a cover shortage, while penalty values less than 100 represent a cover surplus. In the left-hand side nurse roster, there is 100 cover shortage penalty for shift D in $day2$, and the nurse $N1$ is off on this day and has 480 min' available time that are not assigned yet. So, in the right-hand side, the 480 available minutes are utilized by assigning a shift D to the nurse on $day2$ to compensate shortage in shift D on that day.

```

1. Generate  $SH$  set
2. For each  $sh_{s,d} \in SH$ 
   2.1. Create the set  $off_d$ 
   2.2. For each nurse  $n \in off_d$ 
      2.2.1. Calculate  $AT_n$ 
      2.2.2. If  $AT_n \geq$  shift  $s$  duration &  $AS_{n,s} \geq 0$ 
         2.2.2.1. Set  $X_{n,d} = s$ 
         2.2.2.2. Check  $h$ 
         2.2.2.3. If  $h < 0$ 
            2.2.2.3.1. Calculate  $O_c$ 
            2.2.2.3.2. If  $O_c < Best\_O_c$ 
               2.2.2.3.2.1. Set  $Best\_O_c = O_c$ 
               2.2.2.3.2.2. Update  $[X]$ 
            2.2.2.3.3. End if
         2.2.2.4. End if
      2.2.3. End if
      2.2.4. If  $sh_{s,d} = 0$ 
         2.2.4.1. Break the for loop (2.2)
      2.2.5. End if
   2.3. End for
3. End for
4. Return  $[X]$  ,  $O_c$  &  $O_t$ 

```

FIGURE 3. Pseudo code for neighborhood search algorithm of HWCF.1 structure.

Cover penalty table			Cover penalty table		
	day1	day2	day1	day2	day3
<i>E</i>	0	0	<i>E</i>	0	0
<i>D</i>	0	(100)	<i>D</i>	0	0
<i>N</i>	100	0	<i>N</i>	0	1

Nurses roster			AT	Nurses roster			AT
<i>N1</i>			480:min.	<i>N1</i>			0 min.
<i>N2</i>	<i>N</i>	<i>E</i>	0 min.	<i>N2</i>		<i>E</i>	0 min.
<i>N3</i>	<i>D</i>		0 min.	<i>N3</i>		<i>N</i>	0 min.
<i>N4</i>	<i>E</i>	<i>N</i>	0 min.	<i>N4</i>	<i>E</i>	<i>N</i>	0 min.

Before

After

FIGURE 4. Example for HWCF.1 neighborhood search process.

HWCF.2: In this structure, the cover surplus in some shifts is to be used to compensate the cover shortage in other shifts. So, the objective of this structure is two folds, minimizing the cover shortage penalty as well as the cover surplus penalty. The pseudo code in Figure 5 gives a detailed description of the search process using the HWCF.2 structure. An illustrative example is shown in Figure 6. In the left-hand side, there is a cover shortage in shift *D* on *day2*, and a cover surplus in shift *N* on *day3*. The nurse *N1* is *off* on *day2* and has an assignment of shift *N* on *day3*. Therefore, by setting *day3* to be *off*, the resulting available time can be used to assign a shift *D* on *day2*. So, the cover shortage and surplus in both days are eliminated as shown the right-hand side of Figure 6.

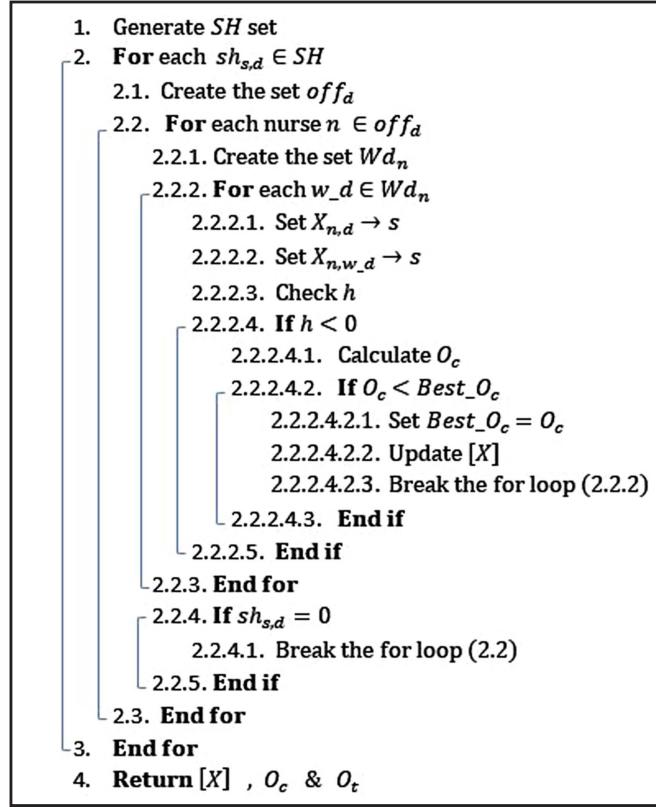


FIGURE 5. Pseudo code for neighborhood search algorithm of HWCF.2 structure.

Cover penalty table			
	day1	day2	day3
<i>E</i>	0	0	0
<i>D</i>	0	(100)	0
<i>N</i>	0	0	(1)

Nurses roster		
<i>N1</i>	<i>N</i>	(<i>N</i>)
<i>N2</i>	<i>N</i>	<i>E</i>
<i>N3</i>	<i>D</i>	<i>N</i>
<i>N4</i>	<i>E</i>	<i>N</i>

Before

Cover penalty table			
	day1	day2	day3
<i>E</i>	0	0	0
<i>D</i>	0	(0)	0
<i>N</i>	0	0	(0)

Nurses roster		
<i>N1</i>	<i>N</i>	(<i>D</i>)
<i>N2</i>	<i>N</i>	<i>D</i>
<i>N3</i>	<i>D</i>	<i>N</i>
<i>N4</i>	<i>E</i>	<i>N</i>

After

FIGURE 6. Example for HWCF.2 neighborhood search process.

4.2.2. Conventional Search structures (CS)

We mean here by conventional structures, these neighborhood structures that take the traditional form of vertical or horizontal swapping/exchanges which are usually used in literature as in [5, 6, 13].

Vertical Swapping structures: These structures include vertical swaps of a single shift or multiple shifts between two different nurses as shown in Figure 7. These swaps are only perturbing the individual schedules of the nurses, without a direct effect on the cover constraint. These swaps contribute in reducing the violations for shifts on/off requests of the nurses. There are six vertical swapping structures to be used here:

	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed
Before	N1	[E]	E	[N]	[L]	[E]		L	L	
	N2		[N]	[N]	E	L	[E]	N	L	
	N3	L		[L]	E		[N]	N		
	N4	[N]	N	E		L	L		N	E
	N5		[E]	[E]	N		L	[N]	E	

	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed
Before	N1	N	E	L	E	N	N	L	L	
	N2		E	E		E	L	N		E
	N3	L			N	L	E		N	
	N4	E	N	E		L	L		N	E
	N5		N	N	N		L		E	N

FIGURE 7. Examples for the vertical swapping structures.

- *VCS.1*; two different single shifts of the same day are swapped vertically between two different nurses.
- *VCS.2*; two blocks of (two consecutive shifts) are swapped vertically between two different nurses.
- *VCS.3*; two blocks of (three consecutive shifts) are swapped vertically between two different nurses.
- *VCS.4*; two blocks of (four consecutive shifts) are swapped vertically between two different nurses.
- *VCS.5*; two blocks of (five consecutive shifts) are swapped vertically between two different nurses.
- *VCS.6*; two weekend blocks (Sat and Sun) are swapped vertically between two different nurses.

Horizontal Swapping structures: In these structures, horizontal swaps are applied for the assignments of a single nurse. These horizontal swaps affect both the cover constraint and the nurses' requests. So, they contribute in improving the total solution penalty as well as perturbing the solution structure. These structures include six different structures as shown in Figure 8.

- *HCS.1*; two different single shifts are swapped horizontally between two different days for the same nurse.
- *HCS.2*; two blocks of (two consecutive days) are swapped horizontally for the same nurse.
- *HCS.3*; two blocks of (three consecutive days) are swapped horizontally for the same nurse.
- *HCS.4*; two blocks of (four consecutive days) are swapped horizontally for the same nurse.
- *HCS.5*; two blocks of (five consecutive days) are swapped horizontally for the same nurse.
- *HCS.6*; two weekend blocks (Sat and Sun) are swapped horizontally for the same nurse.

4.3. The Two-stage search algorithm

In this section, we will describe the two-stage search process and demonstrate how the described neighborhood structures are utilized. In the first stage of the search process, the focus is on minimizing the violations for the cover constraints, especially the cover shortage as it has the highest weight among all of the soft constraints in the problem. The first stage search process is demonstrated by the pseudo code in Figure 9. The algorithm starts by applying the HWCF structures. After that, the vertical swapping structures are applied one by one. For the search diversification purpose, all the feasible swaps are accepted. For each structure, a maximum number of swaps is considered. After each structure, the HWCF group is applied to search for any new possible improvements in the covering constraint violations' penalty. Then, the horizontal swapping structures are applied. Only swaps with cover violations improvement are accepted. After finishing all horizontal

	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed
Before	N1	[E]	E	[N]	L	E		L	L	
	N2	[N]	N	[E]	L			E	N	L
	N3	[L]		L	E	[N]	N	N		
	N4	[N]	N	E	L	L		N	E	
	N5	[E]	E	N		L	N	E		

	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed
After	N1	N	E		E	L	E		L	L
	N2		E	L		N	N		E	N
	N3	N	N	N	E		L		L	
	N4			N	E	L	L	N	N	E
	N5	L	N		E			E	E	N

FIGURE 8. Examples for the horizontal swapping structures.

```

1. Repeat until a termination condition
  1.1. Apply HWCF.1 & HWCF.2
2. End repeat
3. Repeat until a termination condition
  3.1. For  $i = 1:6$ 
    3.1.1. Apply VCS. $i$  → Accept all feasible swaps
    3.1.2. Apply HWCF.1 & HWCF.2
  3.2. End for
  3.3. For  $i = 1:6$ 
    3.3.1. Apply HCS. $i$  → Accept only swaps with  $O_c$  improvement
  3.4. End for
  3.5. Apply HWCF.1 & HWCF.2
4. End repeat
5. Return the best solution (minimum  $O_t$ )

```

FIGURE 9. Pseudo code for the first stage of the VNS algorithm.

structures, the HWCF group is applied again. At the end of the stage, the best solution – with the minimum total penalty – is kept to be used in the second stage.

The second stage considers minimizing the total solution penalty of all soft constraints violations. Indeed, in this stage more focus is considered for the nurses' requests soft constraint – shifts on/off requests. As illustrated in the pseudo code in Figure 10, both vertical and horizontal swaps are used and only the improving swaps are accepted. Therefore, quick improvements can be achieved regarding shifts on/off requests constraint.

5. COMPUTATIONAL RESULTS

The proposed two-stage neighborhood search algorithm was tested on a recent 24 benchmark instances [8]. Table 3 summarizes the main characteristics of these instances. The variation in problem size and complexity of these instances makes them an appropriate benchmark to evaluate the proposed algorithm.

```

1. Repeat until a termination condition
  1.1. For  $i = 1:6$ 
    1.1.1. Apply  $VCS.i \rightarrow$  Accept improving swaps only
  1.2. End for
  1.3. For  $i = 1:6$ 
    1.3.1. Apply  $HCS.i \rightarrow$  Accept improving swaps only
  1.4. End for
  1.5. Apply  $HWCF.1$  &  $HWCF.2$ 
2. End repeat
3. Return the best solution (minimum  $O_t$ )

```

FIGURE 10. Pseudo code for the second stage of the VNS algorithm.

TABLE 3. Benchmark instances main characteristics.

#	Days	Nurses	Shifts	Days-off	Shifts on/off
01	14	8	1	8	26
02	14	14	2	14	62
03	14	20	3	20	64
04	28	10	2	20	71
05	28	16	2	32	106
06	28	18	3	36	135
07	28	20	3	40	168
08	28	30	4	60	225
09	28	36	4	72	232
10	28	40	5	80	284
11	28	50	6	100	336
12	28	60	10	120	422
13	28	120	18	240	841
14	42	32	4	128	359
15	42	45	6	180	490
16	56	20	3	120	280
17	56	30	4	160	480
18	84	22	3	176	414
19	84	40	5	320	834
20	182	50	6	900	2318
21	182	100	8	1800	4702
22	364	50	10	1800	4638
23	364	100	16	3600	9410
24	364	150	32	5400	13 809

The algorithm is coded on (Matlab® R2018b). The experiments were implemented on Intel Core 2 Duo 2 GHz, 4 GB RAM. The total allowed runtime is set to one hour similar to literature. In order to define the best relative length of runtime dedicated for each of the two stages, pilot experiments were performed on four instances with different characteristics, namely *Instance8*, *Instance13*, *Instance18*, and *Instance23*. In these experiments, three different time distribution were tested as follows;

- 30% of time is dedicated for first stage, and 70% for the second stage (30%–70%).
- 50% of time dedicated for first stage, and 50% for the second stage (50%–50%).

TABLE 4. Results for the two stages relative runtime length experiments.

Relative length	Instance 8	Instance 13	Instance 18	Instance 23
(30%–70%)	1542	3103	5926	347 205
(50%–50%)	1443	2712	5526	206 744
(70%–30%)	1499	2783	5537	219 716

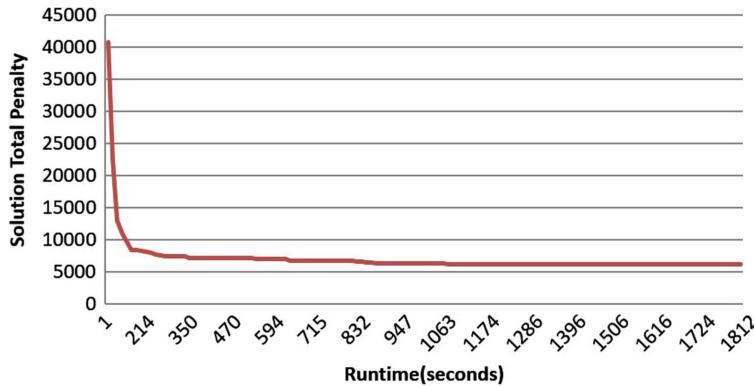


FIGURE 11. A sample convergence curve for the first stage of the Two-stage VNS algorithm.

- 70% of time dedicated for first stage, and 30% for the second stage (70%–30%).

As shown in Table 4, the experiments revealed that the equal distribution (50%–50%) of runtime between the two stages achieves the best solutions quality (least solution penalty) compared to the other two scenarios for all the four tested instances. Therefore, the (50%–50%) setting is used for all instances' experiments.

For the vertical and horizontal swaps, a maximum number of swaps is set for each structure. It is worth mentioning that the value of the maximum number of swaps differs according to the neighborhood structure itself as well as the problem size. As the problem size increases, the number of swaps increases. However, for the same problem size, as the size of the blocks of the structure increases, the number of swaps decreases.

Figures 11 and 12 show example of convergence curves for each stage of the search process independently. These curves demonstrate the improvement of the solution *versus* the runtime (in seconds) for a sample run of *Instance18*. Considering the different scales of the two curves, it is worth mentioning that the first stage has the largest contribution in the overall algorithm convergence to the final solution. Of course, this is rational as the first stage deals with constraints with higher penalty weights. Figure 11 demonstrates the rapid improvement in the solution penalty in the first stage of the algorithm. Thanks to the HWCF structures that enabled such a performance.

For a fair comparison, a similar method is needed. Therefore, the reported results in [8] for one hour runtime of the ejection chain heuristic are used for comparison. For detailed description of the ejection chain heuristic, the reader is referred to [4,5].

Table 5 compares the results of the proposed two-stage variable neighborhood search (VNS) algorithm with the ejection chain heuristic results. As seen in the table, the proposed two-stage VNS outperforms the ejection chain heuristic in 16 instances (shown in bold style), and produces the same result for *Instance01* only. However, it cannot produce better results for seven instances (those with negative improvement). The fourth column shows the percent improvement achieved in the solution compared to the ejection chain heuristic results. The resulting improvements ranged from 0.2% in *Instance02* and reached more than 70% in *Instance22*. Two instances from those seven that the ejection chain method outperforms the proposed algorithm are with only slight differences

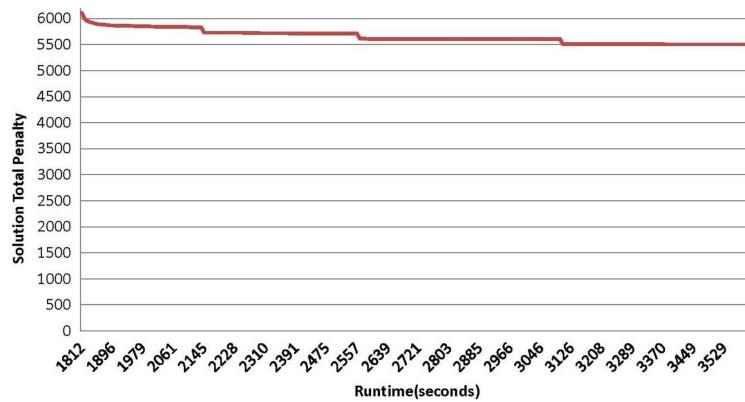


FIGURE 12. A sample convergence curve for the second stage of the Two-stage VNS algorithm.

TABLE 5. The best results of the Two-stage VNS algorithm in comparison with the best results of the Ejection chain heuristic.

#	Ejection Chain	Two-stage VNS	% Improvement
01	607	607	0.0%
02	837	835	0.2%
03	1003	1012	-0.9%
04	1718	1728	-0.6%
05	1358	1257	7%
06	2258	2167	4%
07	1269	1110	13%
08	2260	1443	36%
09	463	456	1.5%
10	4797	4784	0.3%
11	3661	3661	0.0%
12	5211	4344	17%
13	3037	2712	11%
14	1847	1465	21%
15	5935	4838	18%
16	4048	3981	2%
17	7835	6420	18%
18	6404	5526	14%
19	5531	5670	-3%
20	9750	18 876	-94%
21	36 688	58 995	-61%
22	516 686	142 778	72%
23	54 384	206 744	-280%
24	156 858	792 331	-405%

(-0.9% and -0.6%). It is worth mentioning that among the most challenging five instances (*Instance20* to *Instance24*); the proposed two-stage VNS algorithm is able to produce better results for one of them. Indeed, these five instances are considered the most challenging due to their huge size in terms of planning horizon, number of nurses and shift types as well as the large number of day-off requests and shift on/off requests. The reason for the obtained poor-quality solutions for multiple of these huge size instances, mostly due to the exhaustive search process in the first stage that mainly depends on the number of days, number of shift types and the number of nurses. Therefore, due to the huge size of these instances' characteristics, the first stage search process requires longer computational time, and so less number of iterations within the specified time limit which results in poor-quality solutions. By extending the runtime for two of those challenging instances (*instances 23* and *24*), the algorithm is able to provide better results than the ejection chain method in 4 h and 7.5 h, respectively. From a practical perspective, these extended runtimes are still considered a short processing time for such large size instances that are to be processed on annual basis.

6. CONCLUSION

This paper proposed a new two-stage variable neighborhood search algorithm for solving the nurse rostering problem. At the first stage, after generating an initial solution, mainly two new neighborhood structures are employed to improve the solution by focusing on soft constraints with the highest penalty weights. For diversification purpose, a set of vertical and horizontal swaps are applied. After that, the second stage considers improving the overall solution penalty taking into account all the soft constraints of the problem. The proposed algorithm was evaluated using 24 benchmark instances and compared with an ejection chain heuristic approach from literature. The results showed the superiority of the proposed approach over the ejection chain heuristic method for most of the instances.

Future research includes investigating the proposed two new structures of the first stage of the algorithm to find possible improvements that enable them to solve the large size instances more efficiently. Another important point that have to be studied, is the incorporation with other exact or heuristic method.

Acknowledgements. This research project is sponsored by the Egyptian ministry of higher education grant and the Japanese International Cooperation Agency (JICA) in the scope of the Egypt-Japan University of Science and Technology.

REFERENCES

- [1] M. Abdelgalil, Z. Yahia and A.B. Eltawil, A proposed new dynamic programming formulation for the nurse rostering problem. In: *Proceedings of 47th International Conference on Computers and Industrial Engineering* (2017).
- [2] M.A. Awadallah, A.T. Khader, M.A. Al-Betar and A.L. Bolaji, Global best harmony search with a new pitch adjustment designed for nurse rostering. *J. King Saud Univ. Comput. Inf. Sci.* **25** (2013) 145–162.
- [3] M.A. Awadallah, A.L. Bolaji and M.A. Al-Betar, A hybrid artificial bee colony for a nurse rostering problem. *Appl. Soft Comput.* **35** (2015) 726–739.
- [4] E.K. Burke and T. Curtois, New approaches to nurse rostering benchmark instances. *Eur. J. Oper. Res.* **237** (2014) 71–81.
- [5] E.K. Burke, T. Curtois, R. Qu and G. Vanden Berghe, A time predefined variable depth search for nurse rostering. *Informs J. Comput.* **25** (2013) 411–419.
- [6] E.K. Burke, J. Li and R. Qu, A hybrid model of integer programming and variable neighbourhood search for highly-constrained nurse rostering problems. *Eur. J. Oper. Res.* **203** (2010) 484–493.
- [7] S. Ceschia, N.T.T. Dang, P. De Causmaecker, S. Haspeslagh and A. Schaefer, Second International Nurse Rostering Competition (INRC-II) – Problem description and rules. Preprint [arXiv:1501.04177](https://arxiv.org/abs/1501.04177) (2015).
- [8] T. Curtois and R. Qu, *Computational results on new staff scheduling benchmark instances*. Technical report, ASAP Research Group, School of Computer Science, University of Nottingham (2014). http://schedulingbenchmarks.org/papers/computational_results_on_new_staff_scheduling_benchmark_instances.pdf.
- [9] S.C. Gao and C.W. Lin, Particle swarm optimization based nurses shift scheduling. In: *Proceedings of the Institute of Industrial Engineers Asian Conference* (2013) 775–782.
- [10] C.A. Glass and R.A. Knight, The nurse rostering problem: a critical appraisal of the problem structure. *Eur. J. Oper. Res.* **202** (2010) 379–389.
- [11] M. Hadwan, M. Ayob, N.R. Sabar and R. Qu, A harmony search algorithm for nurse rostering problems. *Inf. Sci.* **233** (2013) 126–140.

- [12] S. Haspeslagh, P. De Causmaecker, A. Schaerf and M. Stolevik, The first international nurse rostering competition 2010. *Ann. Oper. Res.* **218** (2014) 221–236.
- [13] S. Martin, D. Ouelhadj, P. Smet, G. Vanden Berghe and E. Zcan, Cooperative search for fair nurse rosters. *Expert Syst. App.* **40** (2013) 6674–6683.
- [14] R. M'Hallah and A. Alkhabbaz, Scheduling of nurses: a case study of a Kuwaiti health care unit. *Oper. Res. Health Care* **2** (2013) 1–19.
- [15] E. Rönnberg, T. Larsson and A. Bertilsson, Automatic scheduling of nurses: What does it take in practice? In: Systems Analysis Tools for Better Health Care Delivery (2013) 151–178.
- [16] I.X. Tassopoulos, I.P. Solos and G.N. Beligiannis, A two-phase adaptive variable neighborhood approach for nurse rostering. *Comput. Oper. Res.* **60** (2015) 150–169.
- [17] C. Valouxis, C. Gogos, G. Goulas, P. Alefragis and E. Housos, A systematic two phase approach for the nurse rostering problem. *Eur. J. Oper. Res.* **219** (2012) 425–433.
- [18] T.C. Wong, M. Xu and K.S. Chin, A two-stage heuristic approach for nurse scheduling problem: a case study in an emergency department. *Comput. Oper. Res.* **51** (2014) 99–110.
- [19] World Health Organization, Nursing and midwifery. <http://www.who.int/mediacentre/factsheets/nursing-midwifery/en/> (2018).
- [20] T.H. Wu, J.Y. Yeh and Y.M. Lee, A particle swarm optimization approach with refinement procedure for nurse rostering problem. *Comput. Oper. Res.* **54** (2015) 52–63.
- [21] Z. Zheng and X. Gong, Chemical reaction optimization for nurse rostering problem. In: Frontier and Future Development of Information Technology in Medicine and Education (2014) 3275–3279.
- [22] L. Zhipeng and J. Hao, Adaptive neighborhood search for nurse rostering. *Eur. J. Oper. Res.* **218** (2012) 865–876.