Ece Cetin Yagmur and Ahmet Sarucan*

Nurse Scheduling with Opposition-Based Parallel Harmony Search Algorithm

<https://doi.org/10.1515/jisys-2017-0150>

Received April 11, 2017; previously published online September 15, 2017.

Abstract: One of the advances made in the management of human resources for the effective implementation of service delivery is the creation of personnel schedules. In this context, especially in terms of the majority of health-care systems, creating nurse schedules comes to the fore. Nurse scheduling problem (NSP) is a complex optimization problem that allows for the preparation of an appropriate schedule for nurses and, in doing so, considers the system constraints such as legal regulations, nurses' preferences, and hospital policies and requirements. There are many studies in the literature that use exact solution algorithms, heuristics, and meta-heuristics approaches. Especially in large-scale problems, for which deterministic methods may require too much time and cost to reach a solution, heuristics and meta-heuristic approaches come to the fore instead of exact methods. In the first phase of the study, harmony search algorithm (HSA), which has shown progress recently and can be adapted to many problems is applied for a dataset in the literature, and the algorithm's performance is evaluated by comparing the results with other heuristics which is applied to the same dataset. As a result of the evaluation, the performance of the classical HSA is inadequate when compared to other heuristics. In the second phase of our study, by considering new approaches proposed by the literature for HSA, the effects on the algorithm's performance of these approaches are investigated and we tried to improve the performance of the algorithm. With the results, it has been determined that the improved algorithm, which is called opposition-based parallel HSA, can be used effectively for NSPs.

Keywords: Harmony search algorithm, nurse scheduling problem, meta-heuristics, parallel grouping, opposition-based learning.

1 Introduction

The most important factor in measuring productivity in the service sector is quality. Because human resource is the main source used in the service sector, effective use of the workforce is of great importance to improve service quality. Health service, which is located within the service sector, is one of the fundamental systems of society and economy. The output of the health-care system is human life. Thus, providing an effective, efficient, and high-quality health service is vital for this sector. To achieve this, adequate quantity of health-care personnel with adequate levels of qualifications should be employed in the right place and at the right time. One of the studies conducted on the effective use of human resources is the creation of personnel schedules. Nursing forms the largest body of employees in the health-care system, spanning all segments of care [15]. From this perspective, nurse scheduling problem (NSP) has been a key subject for researchers and has maintained its importance for many years.

NSP, which deals with the assignment of nurses to shifts under some constraints, is a complex optimization problem that has been studied for >40 years. These constraints consist of hospital requirements, legal restrictions, and nurses' expectations, and they can vary from one problem to another owing to the problem structure. Commonly occurring constraints for NSP are listed by Cheang et al. [9]. Hospital objectives include

*Corresponding author: Ahmet Sarucan, Department of Industrial Engineering, Faculty of Engineering, Selcuk Universitesi, Konya 42050, Turkey, e-mail: sarucan@gmail.com

Ece Cetin Yagmur: Department of Industrial Engineering, Faculty of Engineering, Selcuk Universitesi, Konya, Turkey

ensuring continuous service with appropriate nursing skills and staffing size. While all these are achieved, the knowledge, skills, and capabilities of the workforce should be considered. In practice, schedules are generated based on the demand for nurses by considering legal restrictions, which aim to minimize labor cost in terms of hospital budget. At this point, the biggest noticeable shortfall is that the preferences and wishes of nurses working in shifts have not been sufficiently taken into account. For the purpose of increasing job satisfaction, nurses' preferences and their specific demands about the shift type, working hours, working patterns, requested days off, etc., should also be taken into consideration. When all are examined within this framework, the creation of a schedule becomes a challenging and time-consuming task. Thus, computer-aided approaches have gained more importance over manual approaches for the desired quality schedules.

There have been many solution approaches for NSP in the literature. Although the first studies on NSP suggested traditional mathematical models, these models could not be used effectively today due to the complex nature of real-life problems. Therefore, non-deterministic approaches have attracted the attention of researchers in recent years. Meta-heuristic approaches, which are used for the investigation of the search space effectively with different strategies, have now come to the fore. In this way, optimal or near-optimal solutions can be found in a reasonable computing time.

Osogami and Imai [27] classified NSP as an NP-hard optimization problem and took advantage of the local search approach by using different neighborhood operations. Dias et al. [11] benefited from both genetic algorithms (GAs) and tabu search, and designed a user-friendly interface to select alternative solutions from both algorithms. Aickelin and Li [1] used an evolutionary algorithm, which was called an estimation of the distribution algorithm, to solve the problem, and their study is the first study to apply Bayesian networks to personnel scheduling. Gutjahr and Rauner [13] applied the ant colony optimization approach for NSP for the first time, and compared this algorithm with the greedy earliest assignment strategy. In a study by Li et al. [20], the convergence of the meta-heuristic approach was accelerated with the goal programming approach. Brucker et al. [6] used the greedy local search strategy to improve solution quality. Smet et al. [29] compared their two-phase heuristic with the mixed integer programming solution. Wong et al. [35] suggested a two-stage heuristic approach, which is based on the Excel spreadsheet, for a case study in an emergency department. Tassopoulos et al. [31] used the variable neighborhood search approach for the First International Nurse Rostering Competition (INRC-2010) dataset. Wu et al. [36] benefited from the particle swarm optimization approach by emphasizing the complex nature of the problem. It is seen that, in recent years, both local search and population-based approaches have been successively applied to the NSP and there are many more papers in the literature, which can be seen in more detail in the study by Burke et al. [7].

Although numerous studies have been carried out on NSP, there are a few studies that allowed researchers to compare their algorithms. The necessity of a benchmark dataset for NSP was emphasized both by Cheang et al. [9] and Burke et al. [7] to motivate researchers to reach better solutions for the same problem. Thus, in this study, we used common data, as proposed by Vanhoucke and Maenhout [33]. The problem structure and parameters are given in details in the following section.

Many heuristic algorithms in the literature are generated based on the imitation of events in nature. One of them is the harmony search algorithm (HSA), which was developed by Geem et al. [12]. HSA explores achievement of the best harmony with notes played in an orchestra. It is considered a population-based algorithm with local-based aspects by Lee and Geem [19]. The application areas of the algorithm and the contributions made to the algorithm since its emergence were examined in a study by Ingram and Zhang [16]. According to Kougiass and Theodosiou [18], half of the published HSA studies until 2005 were related to the design of water distribution networks; however, since then, the range of applications has expanded and today HSA has been successfully adapted to many different optimization problems. One of the application areas of HSA is NSP. The first study that tested HSA on NSP was carried out by Awadallah et al. [3], and the performance of their proposed HSA was evaluated using datasets established by INRC-2010. The results produced by the algorithm were not comparatively better than the best results found by competitors. In a study by Hadwan et al. [14], HSA was compared with GA for a case study, and HSA outperformed GA on all tested instances. In addition to

the case study, the researchers conducted a series of experiments to test the efficiency of HSA with different parameter settings on a benchmark problem. Awadallah et al. [4] made two major improvements on the HSA. One of them was the replacement of random selection with the global-best selection of particle swarm optimization while selecting harmony from the memory and the other was the creation of multipitch adjustment procedures. According to the results, they were able to provide their solution quality on the INRC dataset as compared to their previous study in 2011. As seen in the literature, HSA can be considered an effective algorithm for NSP. However, from the studies, the maximum number of improvisations (NI) was thought to be too high. In a study by Awadallah et al. [3], NI was between 100,000 and 300,000, where the search process stops after 5000 iterations without improvement. In a study by Hadwan et al. [14], NI was adjusted to 50,000 iterations and in a study by Awadallah et al. [4] it was adjusted to 100,000. Therefore, to enhance the algorithm performance in terms of time, parallel search has been found useful for HSA. In the literature, several novel variants of the parallel search for HSA have been presented. In their study, Pan et al. [28] presented a local-best HSA with dynamic subpopulations. According to this approach, the whole harmony memory (HM) was divided into many small-sized sub-HMs, and the search was performed in each sub-HM independently. As the best harmony vector often carries better information than others during the evolution process, researchers benefited from good information captured in the local best harmony vector. Askarzadeh and Rezazadeh [2] proposed a grouping-based global HS algorithm. According to their research, HM was divided into three equal groups according to their solution quality and in the process of creating a new harmony, a tournament selection approach was used. We et al. [34] presented a parallel HSA with dynamic HM size. They stressed the need to take full advantage of all of the information hidden in the HM. Therefore, they did not make a grouping in the form of bad and good harmonies.

In their study, Omran and Mahdavi [26] stated that the small-sized HM works better than a large one for the HSA, and according to Liang and Suganthan [21], dynamic subpopulation can effectively balance the fast convergence and large diversity. Motivated by the mentioned claims, we used a parallel searching approach for HSA to solve NSP. In nature-inspired meta-heuristic algorithms, two key components are local intensification and global diversification, and their interaction can significantly affect the efficiency of a meta-heuristic algorithm [37]. In parallel HSA, local intensification is provided by the division of the whole HM into many small-sized sub-HMs. In each iteration, the sub-HMs are regrouped, so global diversification is also guaranteed by the exchange of information between the subgroups. In addition to these, we were inspired by the opposition-based learning approach proposed by Tizhoosh [32] for the coverage of the entire search space. According to our proposed parallel HSA, using the individuals whose penalties are rather far from each other is suitable for the generation of the initial population in terms of overall representation of the search space. Banerjee et al. [5] used this approach for HSA for continuous optimization problems with the opposite number concept. In this study inspired from this approach, both in the phase of initial population generation and in the phase of division of whole HM into sub-HMs, we used the harmonies whose penalty values are far away from each other.

The aim of our study is to examine all aspects of HSA and to evaluate the success of our proposed HSA on a common set of problems in NSP. In addition to this, researches that determined the factors that increase the effectiveness of the improved algorithm are within the scope of our study. According to the results, the performance of the classical HSA algorithm was inadequate when compared with the other heuristics applied to the same problem. Thus, in order to improve the performance of the algorithm, the parallel HSA approach is proposed. To the best of our knowledge, this is the first study to apply parallel HSA for the NSP. The originality of our study lies in the fact that we ensured further development of solution quality by adding a new feature to the parallel HSA.

The remainder of this paper is as follows: In section 2, the characteristics of examined problem are defined. In addition, this section elaborates on a conceptual framework for HSA with the parameter optimization study and presents the classical HSA and our proposed HSA for NSP. In Section 3 the results of the classical HSA, parallel HSA and our proposed HSA are given and these results are compared with not only other heuristics but also the optimal results found by mathematical models. Finally, some conclusions and future research directions are drawn in Section 4.

2 Nurse Scheduling Problem

2.1 Problem Description

Maenhout and Vanhoucke [24] created a data library that is called NSPLIB (Nurse Scheduling Problem Library). The benchmark instances have been grouped in two different sets according to the length of planning horizon: a diverse set and a realistic set. In this research, we used the diverse set, which deals with weekly schedules (7 days) with 25 nurses and 4 shifts, and solved the first 100 problem instances in each case. The generated data set was shaped by three main criteria, as described below.

Preference constraints: The preference matrix contains the information about a particular day or shift at which the nurses want to work. The penalty values range from 1 to 4. As shown in Table 1, if nurse 2 is assigned to morning shift at the first day, the penalty value is 1; if nurse 2 is assigned to other shifts, the penalty value is 2.

Coverage constraints: The coverage matrix expresses the number of nurses needed for a particular day in a particular shift. It can be seen from the Table 2, for example, that the required number of nurses for night shift at the first day should be at least two.

Case-specific constraints: The diverse set consists of eight specific cases, as seen in Table 3. The values in the table are given as minimum and maximum. For example, according to case 7:

- A nurse should work 5 days.
- A nurse should be assigned a maximum of five morning shifts, 5 day shifts, and three night shifts.

Table 1: An Illustration of Preference Matrix.

	Day 1				Day 2			
	Shift 1	Shift 2	Shift 3	Shift 4	Shift 1	Shift 2	Shift 3	Shift 4
Nurse 1	2	4	4	4	1	1	2	1
Nurse 2	1	2	2	2	1	2	3	3
Nurse 3	3	4	4	3	1	3	4	2

Table 2: An Illustration of Coverage Matrix.

	Shift 1 (Morning)	Shift 2 (Day)	Shift 3 (Night)	Shift 4 (Day-off)
Day 1	2	1	2	0
Day 2	4	2	2	0
Day 3	3	5	3	0

Table 3: Case-Specific Constraints.

	Number of assignments	Number of assignments per shift	Consecutive working shifts	Consecutive same working shifts
Case 1	(5,5)	×	×	×
Case 2	(4,6)	×	×	×
Case 3	(5,5)	(1,3), (1,3), (1,2)	×	×
Case 4	(4,5)	(0,5), (0,5), (0,4)	×	×
Case 5	(5,5)	×	(2,5)	×
Case 6	(4,6)	×	(1,5)	×
Case 7	(5,5)	(0,5), (0,5), (0,3)	(2,5)	(2,3), (2,3), (2,3)
Case 8	(2,6)	(0,6), (0,6), (0,3)	(2,4)	(1,4), (1,4), (2,4)

- A nurse should not work more than five consecutive days and less than two consecutive days.
- A nurse should be assigned to the same working shifts a minimum of two and a maximum of three times consecutively for a scheduling period.

In addition to the case-specific constraints, successive assignments between a night and a morning shift, a night and a day shift, and a day and a morning shift were forbidden. Thus, there must be at least 24 h between consecutive shifts. More detailed information about the NSP generator can be found in Vanhoucke and Maenhout [33].

The constraints of the discussed problem can be classified into two main groups. While coverage constraints and case-specific constraints are thought to be hard constraints, the preference of nurses can be considered as soft constraints. Thus, the objective of the problem is to find a feasible nurse schedule that satisfies all hard constraints, so that the total penalty costs caused by soft constraints are minimized.

The NSP has been formulated from a shift-pattern view by Maenhout and Vanhoucke [24], and the researchers referred to their previous simple integer programming model, which contains 10 nurse, 3 shifts, and 7 days. For the purpose of evaluating our algorithm performance against the global optimal solution, we benefited from mathematical programming, which was solved by the GAMS package program. Thus, our results are compared not only with other heuristics but also with the optimal results found by the mathematical model.

2.2 HSA for NSP

HSA is a new population-based meta-heuristic algorithm that was proposed by Geem et al. [12]. The pseudo-code using the following parameters of HSA is shown in Figure 1.

The parameters that affected the performance of HSA are as follows:

- HM size (HMS), which represents the number of solution vectors in HM.
- HM consideration rate (HMCR), which is used to determine whether the value of a decision variable will be selected from HM or randomly from the possible range of the considered decision variable.

Harmony Search Algorithm

```

begin
  define objective function  $f(x)$ ,  $x = (x_1, x_2, \dots, x_N)^T$ 
  set  $HMS$ ,  $HMCR$ ,  $PAR$ ,  $NI$ 
  initialize the  $HM$ 
for  $i = 1$  to maximum number of iterations ( $NI$ ) do
   $x^i = \text{empty}$ 
  for  $j = 1$  to number of decision variables ( $N$ ) do
    if ( $\text{rand} \in (0,1) \leq HMCR$ ) then
      choose a value from  $HM$  for  $x_j$ 
      if ( $\text{rand} \in (0,1) \leq PAR$ ) then
        adjust the value of  $x_j$  by:
         $x_{j\text{ new}} = x_{j\text{ old}} \pm bw$ 
      end if
    else
      choose a random variable for  $x_j$  from the possible range of the considered decision
      variable
    end if
  end for  $j$ 
  evaluate new harmony
  if ( $\text{objective function}(\text{new harmony}) \leq \text{worst}(\text{objective function}(HM))$ ) then
    accept the new harmony
  end if
end for  $i$ 
  find the current best solution
end

```

Figure 1: Pseudo-code of the HSA.

- Pitch adjusting rate (PAR), which is used to adjust the value of the decision variable selected from HM by either increasing or decreasing its value with respect to its upper and lower bounds.
- NI, which is used for the stopping criteria.

On the basis of the procedure described above, implementation of HSA for NSP can be summarized as follows:

Step 1. Setting the HSA parameters and problem parameters

In this step, we used factorial analysis of variance to determine the optimum parameters. The data were obtained from the first 30 problems selected from case 1 and the algorithm was ran three times for each problem.

- HMCR effect: Khadwilard et al. [17] claimed that if the HMCR is too low, only a few best harmonies are selected and they may converge too slowly. If this rate is extremely high, almost all the harmonies are used in the HM and then other harmonies are not explored well, leading to potentially wrong solutions. Thus, in this study, the HMCR examined was between 0.8 and 0.99. Furthermore, the dynamic approach, which was proposed by Daham et al. [10], is also taken into consideration. According to this approach, the HMCR rate changes dynamically in each iteration based on Eq. (1):

$$HMCR_{iter} = HMCR_{min} + \frac{HMCR_{max} - HMCR_{min}}{maxiter} * iter \quad (1)$$

where $HMCR_{min} = 0.8$ $HMCR_{max} = 0.99$

- PAR effect: The pitch adjustment operator is similar to the mutation operator in GAs. Sriharsha and Reddy [30] claimed that a low PAR can slow down the convergence of HS because of the limitation in the exploration of only a small subspace of the whole search space. On the other hand, a very high PAR may cause the solution to scatter around some potential optima as in a random search. According to this operator, the shift pattern (includes 7 days) of selected nurses will be exchanged with another random nurse's shift pattern with probability PAR. In this study, we used PAR between 0.01 and 0.2.
- HMS effect: In computational evolution, the population size is directly related to the search space and computation overhead [22]. A small population size with a large iteration cannot explore the whole search space effectively despite the fact that a large population size could make the algorithm expend more computation time. Therefore, the ideal HMS size varies according to the number of maximum iterations. In this study, we examined HM that includes 1, 50, or 100 individuals for 5000 iterations.

In light of this information, factors and their levels used in multifactor analyses of variance are listed in Table 4.

Consequently, the results of the full factorial experimental design with 4320 solutions including $3*4*4*3*30$ (level of HMS * level of PAR * level of HMCR * run times * problem size) are given in Table 5.

Table 4: Experimental Factors and Their Levels.

Factors	Levels	<i>N</i>
HMS	1	48
	50	48
	100	48
PAR	0.01	36
	0.05	36
	0.1	36
	0.2	36
	Dynamic	36
HMCR	0.8	36
	0.9	36
	0.99	36
	Dynamic	36

Table 5: Analysis of Variance on HSA.

	DF	Seq SS	Adj SS	Adj MS	F	p-Value
HMS	2	668.11	668.11	334.06	1084.34	0.000
HMCR	3	3550.79	3550.79	1183.6	3841.93	0.000
PAR	3	7626.04	7626.04	2542.01	8251.33	0.000
HMS * HMCR	6	263.76	263.76	43.96	142.69	0.000
HMS * PAR	6	181.36	181.36	30.23	98.11	0.000
HMCR * PAR	9	220.39	220.39	24.49	79.49	0.000
HMS * HMCR * PAR	18	71.73	71.73	3.98	12.93	0.000
Error	96	29.58	29.58	0.31		
Total	143	12,611.74				

$S = 0.555043$, $R^2 = 99.77\%$, $R^2(\text{adj}) = 99.65\%$.

From Table 5, it can be seen that all HSA parameters including HMS, PAR, and HMCR were statistically significant with 95% confidence interval ($p < 0.05$). In Figure 2, the main effects of factors are illustrated. In addition to this, the interaction pilots are given in Figure 3.

When an interaction is significant, we focus our attention on the interaction, not on the main effects. Because the F-test for the interaction is significant, we benefited from the application of a post hoc procedure to explore which pairs of cell means are significantly different. The results for a three-way interaction using the Tukey procedure revealed that no significant difference was found for HMS at 100, HMCR at dynamic, PAR at 0.01; HMS at 50, HMCR at dynamic, PAR at 0.01; and HMS at 100, HMCR at 0.99, PAR at 0.05, and these combinations are the best choices considering minimization of the penalty value. Thus, it can be summarized that HMS/NI at 50/5000, HMCR at dynamic, and PAR at 0.01 were one of the best settings for our problem, and we used these parameters.

Step 2. Initializing the HM

HM is a matrix that incorporates as many feasible solutions as HMS. It should be particularly emphasized at this stage that due to the large size of the solution space, performing a full schedule creation process in two stages would be more effective. Therefore, in the first phase of our study, we created feasible individual schedules for each nurse; then, in the second phase, we provided an appropriate schedule also considering coverage constraints.

As mentioned before, there are eight different cases in the examined problem. The feasible shift patterns providing all the case-specific constraints were used to create individual schedules at the first phase. These patterns were filtered with string functions in C#. The numbers of feasible shift patterns for all cases are listed below (Table 6).

Figure 2: Main Effects Plot of HSA.

Figure 3: Interaction Plots of Main Effects.

The solution vectors of HM are formed by combining the feasible patterns selected randomly for each of the 25 nurses. In this way, HM is filled with as many sets of solution vectors as HMS. Another hard constraint related to the coverage constraint was not taken into consideration in this step. The control of these coverage constraints is ensured by taking a higher penalty value than the soft constraints in case of any deviation from these constraints.

Step 3. Improvising a new harmony

In this step, a new harmony vector will be improvised based on three operators: memory consideration, pitch adjustment, and random selection. Thus, a nurse is assigned to

- A shift pattern randomly from HM with HMCR probability (memory consideration);
- A random shift pattern from the set of all possible patterns with probability $(1 - \text{HMCR})$ (random consideration); or
- The shift pattern of a selected nurse assigned by memory consideration will be exchanged with another random nurse's shift pattern with probability PAR (pitch adjustment).

This process is repeated until a complete schedule is created.

Step 4. Updating the HM

If the penalty value of a new created schedule is better than the worst one in the current HM, then the new solution is included in HM and the worst solution is excluded from HM. Otherwise, a new solution is discarded.

Step 5. Checking the stopping criteria

All these processes are repeated until we reach the maximum number of improvisations. In our study, 5000 iterations are used to compare HSA with other heuristic algorithms, which solved the same problem.

Table 6: Number of Feasible Shift Pattern for All Cases.

Case	Number of shift pattern (Includes 7 Days)
Case 1	1287
Case 2	3114
Case 3	504
Case 4	2631
Case 5	423
Case 6	3058
Case 7	66
Case 8	449

Table 7: Computational Results for Classical HSA.

	Classical HSA		Best-known solutions		GAP%
	Average penalty	Average CPU	Average penalty	Average CPU	
Case 1	310.44	2.79	299.08	1.47	0.0380
Case 2	313.92	2.83	294.54	2.63	0.0658
Case 3	325.15	2.82	317.84	1.93	0.0230
Case 4	313.31	2.81	298.43	1.85	0.0499
Case 5	315.64	2.76	305.80	1.25	0.0322
Case 6	309.76	2.81	294.93	2.52	0.0503
Case 7	317.45	2.60	314.42	2.13	0.0096
Case 8	311.59	2.78	300.58	1.46	0.0366

The algorithms used for comparison include

- A scatter search algorithm [23];
- An electromagnetic meta-heuristic algorithm [24];
- A GA with hybridization of crossover operators [25].

The first 100 problem instances in each case are solved by the HSA with three runs.

From Table 7, we can see the comparison of HSA with the best-known solutions obtained by the three heuristic algorithms mentioned above. The results display the average penalty value for 100 problems and three runs.

According to the results, the classical HSA was able to find a feasible solution that provides all hard constraints for all problems in each case. However, the GAP values showed that HSA was inefficient in terms of the preference's penalty values compared to other heuristics in 5000 iterations.

2.3 Parallel HSA for NSP

As mentioned previously, to enhance the algorithm performance in terms of time, parallel searching has been found useful for HSA. In this study, we have adapted the parallel HSA that was proposed by We et al. [34] for NSP. According to this approach, as seen in Figure 4, the whole HM has been equally divided into sub-HMs and each sub-HM uses its own members to reach better solutions. After the improvisation process, the best remaining harmonies are combined again to form a whole HM for a re-grouping process. Then, each sub-HM was regrouped randomly and restarted on its own search. This process is continued until the stopping criterion is met.

For our problem, we set the HMS at 51 with three sub-HMs. These sub-HMs carry out the improvisation and updating processes of the classical HSA independently.

Figure 4: Parallel Grouping Search [34].

2.4 Proposed Parallel HSA for NSP

In this study, we have developed an effective HSA by adding a new feature to the parallel HSA. The flowchart of our proposed algorithm is shown in Figure 5. Based on this figure, the details of our proposed algorithm [8] are described below.

Step 1: Based on the results of parameter optimization study, we set HSA parameters as follows:

HMS: 51 (three sub-HMs), HMCR: dynamic rate ($\text{HMCR}_{\min} = 0.8$; $\text{HMCR}_{\max} = 0.99$), PAR: 0.01, and NI: 5000.

Step 2: Firstly, virtual HM is filled with sets of solution vectors as many as 1000. The solution vectors, which represent a full schedule, are formed by combining the feasible patterns selected randomly for each of the 25 nurses. These vectors are sorted in an ascending order according to their penalty values. Then, the initial HM is created by choosing the best 30 and the worst 21 harmonies, which are located in virtual HM.

Step 3: The initial HM has been divided into sub-HMs with the same size of good and bad harmonies (10 good, seven bad), and for each sub-HM, step 4 and step 5 below are performed.

Step 4: In the improvisation process, each sub-HM develops independently and a new harmony vector for each subgroup is generated based on three HSA operators (memory consideration, random consideration, and pitch adjustment), as previously reported.

Step 5: The updating process of each sub-HM is independent of one another. If the objective function of the new harmony is better than the worst harmony of the sub-HM, the worst harmony is replaced with the newly created harmony.

Step 6: If the termination criterion is reached, return to the best harmony vector from the whole HM thus far; otherwise, the sub-best harmonies in each sub-HM are selected, and according to these harmonies, step 3–step 5 are repeated until the termination criterion is reached.

3 Comparison of the Proposed HSA, Parallel HSA, and Classical HSA

As seen from Table 8, the solutions obtained with the classical HSA were not found to be satisfactory.

By searching subregions of the solution space simultaneously, the exploitation ability of the algorithm increases. Thus, parallel HSA is more successful than classical HSA for all cases, which can be seen in Table 9.

As previously mentioned, the initial population is generated randomly by parallel HSA; however, in our proposed algorithm, we used individuals that are away from each other in this step. Thus, a wider region of the solution space was represented. In addition to this, we benefit from the same approach when dividing the subgroup step. In this way, overlapping of individuals that have the same or very close penalty value in the same subgroup was prevented; therefore, the possibility of recovery has been increased. Table 10 includes the results obtained by the proposed parallel HSA approach.

With these results, we can see that our proposed algorithm is superior to the other two approaches. If we compare this algorithm by the best-known solutions in the literature, it appears that the gap is rather

Figure 5: Flowchart of the Proposed Parallel HSA [8].

low. From this point of view, the proposed algorithm can be considered as an effective method for NSP. In the majority of the solution, we can reach the best-known solutions obtained by other heuristics in a very short time.

Table 8: Result of 5000 Iterations Obtained with Classical HSA.

	Classical HSA		Parallel HSA	
	Average penalty	Average CPU	Average penalty	Average CPU
Case 1	310.44	2.79	304.75	3.89
Case 2	313.92	2.83	302.26	4.27
Case 3	325.15	2.82	321.04	3.34
Case 4	313.31	2.81	306.42	3.98
Case 5	315.64	2.76	310.25	3.24
Case 6	309.76	2.81	302.36	4.11
Case 7	317.45	2.60	315.35	3.13
Case 8	311.59	2.78	304.87	3.45

Table 9: Result of 5000 Iterations Obtained with Parallel HSA.

	Parallel HSA		Proposed HSA	
	Average penalty	Average CPU	Average penalty	Average CPU
Case 1	304.75	3.89	302.12	4.19
Case 2	302.26	4.27	300.18	5.23
Case 3	321.04	3.34	319.56	4.44
Case 4	306.42	3.98	303.56	4.77
Case 5	310.25	3.24	308.07	4.20
Case 6	302.36	4.11	299.74	5.24
Case 7	315.35	3.13	314.75	4.01
Case 8	304.87	3.45	302.92	4.56

Table 10: Result of 5000 Iterations Obtained with Proposed Parallel HSA.

	Proposed HSA	Best-known solutions	GAP%
	Average penalty	Average penalty	
Case 1	302.12	299.08	0.0101
Case 2	300.18	294.54	0.0191
Case 3	319.56	317.84	0.0054
Case 4	303.56	298.43	0.0171
Case 5	308.07	305.80	0.0074
Case 6	299.74	294.93	0.0163
Case 7	314.75	314.42	0.0010
Case 8	302.92	300.58	0.0077

Table 11: Comparison between the Proposed Algorithm and Optimal Solutions.

	Proposed HSA	Optimal solutions	GAP%
	Average penalty	Average penalty	
Case 1	302.12	299.07	0.0101
Case 2	300.18	294.54	0.0191
Case 3	319.56	317.82	0.0054
Case 4	303.56	298.41	0.0172
Case 5	308.07	305.77	0.0075
Case 6	299.74	294.93	0.0163
Case 7	314.75	314.30	0.0014
Case 8	302.92	300.49	0.0080

After this comparison, we found the optimal solutions for these problems by using the GAMS package program, and we see that our algorithm has found the optimal or near-optimal solutions for almost all problems, as seen from Table 11. Thus, this approach has proven its success.

4 Conclusion and Future Research

In real-world applications, when preparing schedules in hospital units run with a shift system, legal restrictions and requests of hospital management must be fully covered. The demand and expectations of the nurses working in the unit should be met to ensure an efficient working environment. Another issue of vital importance in the preparation of schedules is that hospital demand must be satisfied with adequate equipment and personnel. With these problem features, the problem set solved in this study reflects the expectations of many hospitals serving today.

In the first part of the study, we evaluated the results of the classical HSA approach. The obtained results have not been found competitive when compared with the three different heuristic methods for this set of problem. When examining the studies in the literature that are able to reach a successful solution with the classical HSA for NSP, it has been observed that the maximum number of iteration is too high. While Hadwan et al. [14] determined 50,000 iterations as the stopping criterion, Awadallah et al. [3] set this criterion between 100,000 and 300,000, where the search process stops after 5000 iterations without improvement. In terms of comparison with other heuristic approaches that are applied to the same problem, in this study the number of improvisation is set as 5000 iterations as in the other heuristics. For the purpose of increasing the efficiency of this algorithm, the effects of the algorithm parameters are evaluated on 30 different problems from case 1.

In the second part of the study, we proposed the parallel HSA approach by making use of current approaches in the literature. The study is important, as it is the first study carried out using the parallel HSA approach for NSP. According to the result, the parallel HSA approach has achieved better results than the conventional HSA in all eight different situations.

In the following part of the study, when generating the initial population step, it was found appropriate to use individuals whose penalty values are far away from each other, as inspired by opposition-based learning. The same strategy has also been followed for dividing HM into sub-HMs. Thus, the proposed parallel HSA approach is separated from other studies and has gained a different perspective. In the majority of the problems, the proposed algorithm reached the best solutions, which have also been found by other heuristics in a very short time, expressed in seconds. Another reason for producing efficient solutions in a short time is that the problem was examined with a 7 days shift pattern. Thus, the solution space has been reduced considerably, and we have shown that the proposed parallel HSA approach can compete with other heuristics for NSP.

With this study, we aimed to show how the algorithm parameters affect HSA and used the parallel HSA for NSP for the first time. This study can be considered as an initial study that can pave the way for further studies on this issue. Also, in this study, a new feature that is based on opposition-based learning has been added to the parallel HSA, so the algorithm was improved.

In further studies, the algorithm's performance and efficiency can be increased by combining parallel HSA with local search strategies, and the proposed algorithm can be used for solving more complicated problems for NSP or other optimization problems.

Bibliography

- [1] U. Aickelin and J. Li, An estimation of distribution algorithm for nurse scheduling, *Ann. Oper. Res.* **155** (2007), 289–309.
- [2] A. Askarzadeh and A. Rezazadeh, A grouping-based global harmony search algorithm for modeling of proton exchange membrane fuel cell, *Int. J. Hydrogen Energy* **36** (2011), 5047–5053.
- [3] M. A. Awadallah, A. T. Khader, M. A. Al-Betar and A. L. A. Bolaji, *Nurse Scheduling Using Harmony Search*, paper presented at the 2011 Sixth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA), 2011.

- [4] M. A. Awadallah, A. T. Khader, M. A. Al-Betar and A. L. A. Bolaji, Global best harmony search with a new pitch adjustment designed for nurse rostering, *J. King. Saud. Univ. Sci. Comp. and Inform. Sci. (JKSUS)* **25** (2013), 145–162.
- [5] A. Banerjee, V. Mukherjee and S. Ghoshal, An opposition-based harmony search algorithm for engineering optimization problems, *Ain Shams Eng. J.* **5** (2014), 85–101.
- [6] P. Brucker, E. K. Burke, T. Curtois, R. Qu and G. V. Berghe, A shift sequence based approach for nurse scheduling and a new benchmark dataset, *J. Heuristics* **16** (2010), 559–573.
- [7] E. K. Burke, P. De Causmaecker, G. V. Berghe and H. Van Landeghem, The state of the art of nurse rostering, *J. Sched.* **7** (2004), 441–499.
- [8] E. Cetin, A metaheuristic approach for nurse scheduling problem, unpublished masters thesis, University of Selcuk, Konya, 2015.
- [9] B. Cheang, H. Li, A. Lim and B. Rodrigues, Nurse rostering problems – a bibliographic survey, *Eur. J. Oper. Res.* **151** (2003), 447–460.
- [10] B. F. A. Daham, M. N. Mohammed and K. S. Mohammed, Parameter controlled harmony search algorithm for solving the four-color mapping problem, *Int. J. Comput. Inf. Technol.* **3** (2014), 1398–1402.
- [11] T. M. Dias, D. F. Ferber, C. C. De Souza and A. V. Moura, Constructing nurse schedules at large hospitals, *Int. Trans. Oper. Res.* **10** (2003), 245–265.
- [12] Z. W. Geem, J. H. Kim and G. Loganathan, A new heuristic optimization algorithm: harmony search, *Simulation* **76** (2001), 60–68.
- [13] W. J. Gutjahr and M. S. Rauner, An ACO algorithm for a dynamic regional nurse-scheduling problem in Austria, *Comput. Oper. Res.* **34** (2007), 642–666.
- [14] M. Hadwan, M. Ayob, N. R. Sabar and R. Qu, A harmony search algorithm for nurse rostering problems, *Inform. Sci.* **233** (2013), 126–140.
- [15] P. Hogan, L. Moxham and T. Dwyer, Human resource management strategies for the retention of nurses in acute care settings in hospitals in Australia, *Contemp. Nurse* **24** (2007), 189–199.
- [16] G. Ingram and T. Zhang, Overview of applications and developments in the harmony search algorithm, *Music-Inspired Harmony Search Algorithm*, pp. 15–37, Part of the Studies in Computational Intelligence book series (SCI, volume 191), Springer, Berlin-Heidelberg, 2009.
- [17] A. Khadwilard, P. Luangpaiboon and P. Pongcharoen, Full factorial experimental design for parameters selection of harmony search algorithm, *J. Ind. Technol.* **8** (2012), 1–10.
- [18] I. Kougiyas and N. Theodossiou, *A New Music-Inspired Harmony Based Optimization Algorithm. Application in Water Resources Management Problems*, paper presented at the Proceedings of the PRE X International Conference, Corfu, Greece, 2010.
- [19] K. S. Lee and Z. W. Geem, A new structural optimization method based on the harmony search algorithm, *Comput. Struct.* **82** (2004), 781–798.
- [20] J. Li, E. K. Burke, T. Curtois, S. Petrovic and R. Qu, The falling tide algorithm: a new multi-objective approach for complex workforce scheduling, *Omega* **40** (2012), 283–293.
- [21] J. -J. Liang and P. N. Suganthan, *Dynamic Multi-swarm Particle Swarm Optimizer with Local Search*, paper presented at the 2005 IEEE Congress on Evolutionary Computation, 2005.
- [22] F. G. Lobo and C. F. Lima, A review of adaptive population sizing schemes in genetic algorithms, in: *Proceedings of the 7th Annual Workshop on Genetic and Evolutionary Computation*, pp. 228–234, ACM, 2005.
- [23] B. Maenhout and M. Vanhoucke, *New Computational Results for the Nurse Scheduling Problem: A Scatter Search Algorithm*, paper presented at the European Conference on Evolutionary Computation in Combinatorial Optimization, 2006.
- [24] B. Maenhout and M. Vanhoucke, An electromagnetic meta-heuristic for the nurse scheduling problem, *J. Heuristics* **13** (2007), 359–385.
- [25] B. Maenhout and M. Vanhoucke, Comparison and hybridization of crossover operators for the nurse scheduling problem, *Ann. Oper. Res.* **159** (2008), 333–353.
- [26] M. G. Omran and M. Mahdavi, Global-best harmony search, *Appl. Math. Comput.* **198** (2008), 643–656.
- [27] T. Osogami and H. Imai, *Classification of Various Neighborhood Operations for the Nurse Scheduling Problem*, paper presented at the International Symposium on Algorithms and Computation, 2000.
- [28] Q. -K. Pan, P. Suganthan, J. Liang and M. F. Tasgetiren, A local-best harmony search algorithm with dynamic subpopulations, *Eng. Optimiz.* **42** (2010), 101–117.
- [29] P. Smet, T. Wauters, M. Mihaylov and G. V. Berghe, The shift minimisation personnel task scheduling problem: a new hybrid approach and computational insights, *Omega* **46** (2014), 64–73.
- [30] A. Sriharsha and A. R. M. Reddy, Music inspired HS algorithm for determining software design patterns, *Issues* **1** (2014), 230–238.
- [31] 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.
- [32] H. R. Tizhoosh, *Opposition-Based Learning: A New Scheme for Machine Intelligence*, paper presented at the CIMCA/IAW-TIC, 2005.

- [33] M. Vanhoucke and B. Maenhout, *Characterisation and Generation of Nurse Scheduling Problem Instances*, working paper 2005, Ghent University. Available at: wps-feb.ugent.be/Papers/wp_05_339.pdf. Accessed 2015.
- [34] J. We, W. Jing, W. Wei, C. Liulin and J. Qibing, *A Parallel Harmony Search Algorithm with Dynamic Harmony-Memory Size*, paper presented at the 2013 25th Chinese Control and Decision Conference (CCDC), 2013.
- [35] T. Wong, M. Xu and K. Chin, A two-stage heuristic approach for nurse scheduling problem: a case study in an emergency department, *Comput. Oper. Res.* **51** (2014), 99–110.
- [36] 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.
- [37] X. -S. Yang, S. Deb and S. Fong, Metaheuristic algorithms: optimal balance of intensification and diversification, *Appl. Math. Inform. Sci.* **8** (2014), 977.