Journal of Applied Sciences

ISSN 1812-5654

Journal of Applied Sciences 13 (6): 846-853, 2013
ISSN 1812-5654 / DOI: 10.3923/jas.2013.846.853

Enhanced Harmony Search Algorithm for Nurse Rostering Problems

[1]Masri Ayob, [2]Mohammed Hadwan, [1]Mohd. Zakree Ahmad Nazri and [3]Zulkifli Ahmad
[1]Data Mining and Optimization Research Group, Center of Artificial Intelligence Technology, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, 43600, Bangi, Selangor, Malaysia
[2]Department of Computer Science, Faculty of Applied Science, Taiz University, P.O. Box 6314, Taiz, Republic of Yemen
[3]School of Linguistics and Language Studies, Faculty of Social Sciences and Humanities, Universiti Kebangsaan Malaysia, 43600, Bangi, Selangor, Malaysia

Abstract: Drawing up the nurses' duty roster is one of the main key issues that are faced by hospital managements, this study focuses on Nurse Rostering Problem (NRP), an NP-hard problem, that is difficult to solve for optimality. Harmony Search Algorithm (HSA) refers to the meta-heuristic algorithm inspired by the improvisation of Jazz musicians. Due to the problem of slow convergence of the basic HSA, this study attempted to enhance basic HSA (called EHSA). This is done by using a semi cyclic shift patterns in the initialization step to generate the initial harmonies (population) rather than using a fully random mechanism in basic HSA. Furthermore, a dynamic mechanism was employed in EHSA to update the parameter values of harmony memory considering rate and pitch adjusting rate instead of fixed values in basic HSA. A real world dataset from large hospital in Malaysia was used to evaluate the performance of EHSA. Results showed that EHSA can produce high quality rosters in shorter execution time compared to basic HSA. A comparison between EHSA and Adaptive Harmony Search (AHS) is also presented to demonstrate the performance of the proposed method. Better results have been obtained by EHSA compared to AHS.

Key words: Harmony search algorithm, nurse rostering problem, meta-heuristic, semi-cyclic, timetabling

INTRODUCTION

Within the area of healthcare there is a worldwide shortage in nursing personals; a phenomenon known as the "crisis in nursing". High workloads, inflexible schedules and unfair distribution of duty shift slots are among the key issues that cause nurses' job dissatisfaction. This dissatisfaction has a negative impact on the personal life of the staff nurses resulting in high turnover and low rate of nursing recruitment (Burke *et al*., 2004). Consequently, the overall quality of patient care is badly affected. Nurse Rostering Problem (NRP) is an NP-hard combinatorial optimization problems. It involves the task to assign nurses to be on duty shifts for specific period of time, specifying the days on and off duty for each nurse (Glass and Knight, 2009). The constraints that accompany NRPs are divided into types known as hard and soft constraints. Hard constraints represent the mandatory rules and regulations that must be fulfilled. Whereas soft constraints can be violated (if necessary) and are used to evaluate the quality of the generated roster. That is, the more soft constraints are satisfied, the better the quality of the roster. Though it does not promise reaching the optimal solution, approximation approaches are widely adopted to solve NRP.

The NRPs are well known to be a fully-constrained problem and some researchers considered it to be an over-constrained problem (Thornton and Sattar, 1997; Bilgin *et al*., 2012). A wide range of real-world NRPs have been introduced by many researchers in the literature of timetabling (Burke *et al*., 2004). Complex NRPs are considered to be among of the NP-hard problems (Burke *et al*., 2004). The difficulty of NRPs is because of the wide range of the associated hard constraints and soft constraints that sometimes conflicting with each other (Burke *et al*., 2004; Cheang *et al*., 2003). Many meta-heuristic approaches have been used to tackle NRP (in general) due to its ability to cope with various constraints and can generate good solutions in a reasonable execution time. Example of these approaches

Corresponding Author: Masri Ayob, Data Mining and Optimization Research Group, Center of Artificial Intelligence Technology, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, 43600, Bangi, Selangor, Malaysia Tel: +603 8921 6741 Fax: +603 8921 6184

are genetic algorithms (Aickelin and Dowsland, 2004), scatter search algorithm (Burke *et al*., 2010), ant colony optimization (Gutjahr and Rauner, 2007) and improved squeaky wheel optimization (Aickelin *et al*., 2009).

According to Talbi (2009) meta-heuristics can be defined as upper level general methodologies which used as guiding strategies in designing underlying heuristics to tackle optimization problems. Meta-heuristic approaches are generally categorized into two main methods. The first one is local search methods which are very fast in finding better quality solutions but only in local region where it starts the searching without global view of other solutions in the search space (Blum *et al*., 2008). The second one is population-based methods, which are better in identifying the location of good solutions by exploring and capturing a global picture of the search space. However, they are usually not efficient in exploiting the desired solutions in the search space as they focus more in exploration (Blum *et al*., 2008). Therefore, the hybridization between population methods and local search methods to create a right balance between exploration and exploitation is highly recommended (Talbi, 2009; Blum *et al*., 2011). In most respects, these characteristics are found in a new algorithm called Harmony Search Algorithm (HSA) which is a population based method that has some components of local search methods.

HSA is a meta-heuristic algorithm inspired by the improvisation of Jazz musicians (Geem *et al*., 2001). In music performance, each musical player plays one musical note each time and those musical notes combine together to form a harmony. In the optimization process, each variable has a value at each time and those values all together form a solution vector (Ingram and Zhang, 2009). HSA uses stochastic random search that depends on the Harmony Memory Considering Rate (HMCR) and Pitch Adjusting Rate (PAR). HSA combines different characteristics of existing meta-heuristic algorithms. By using the harmony memory, HSA is similar to tabu search algorithm in the ability of preserving the history of previous vectors. Further, it is similar to genetic algorithm in the ability of managing several solution vectors simultaneously. The dynamic change of the parameters of HSA during the run acts is similar to simulated annealing algorithm (Lee and Geem, 2004).

HSA has proved its ability to solve many difficult optimization problems (Ingram and Zhang, 2009) such as university course timetabling (Al-Betar and Khader, 2010), vehicle routing (Geem *et al*., 2005) and other optimization problems (Ingram and Zhang, 2009). However, the problem of slow convergence has been noticed in most of the existing works. Thus, the improvement of the basic HSA has attracted many researchers. Among the earlier studies that attempted to enhance the performance of basic HSA is performed by Mahdavi *et al*. (2007), which proposed to use a dynamic mechanism to change the parameter values of PAR and the Bandwidth (BW) instead of using a fixed PAR and BW. The proposed algorithm is called Improved Harmony Search (IHS) and it is inspired by many of the following variants of HSA Mahdavi *et al*. (2007). Another modification to HSA is done by Omran and Mahdavi (2008), where they have developed another variant of HSA called Global best Harmony Search (GHS) inspired by the idea of swarm intelligence. The GHS differs from IHS in the step of improvisation where they modified the pitch adjustment rule as follows: improvising new solution is affected by the best harmony in the harmony memory instead of moving to the neighbor of current value. The BW is removed in GHS (Omran and Mahdavi, 2008). In study of Coelho and Mariani (2009) another HSA variant that included the grade of the solution vectors is reported. Most of the proposed modification in the state of the art of HSA was directed to enhance the improvisation step. Another essential modification to the improvisation step were introduced by Saka and Hasancebi (2009) and Hasancebi *et al*. (2010) to change the parameter values of HMCR and PAR instead of changing the values of PAR and BW during the algorithm running. Less attention has been paid to improve the initialization step where the initial harmonies generated and filled up randomly. Few attempts were reported to improve the step of initializing the harmony memory. In Degertekin (2008) modified HM initialization step by generating solution vectors twice as the HMS then filling up the HM with the best solution vectors. Wang and Huang (2010) made another attempted of using low-discrepancy sequences instead of random mechanism.

Thus, for the purpose of enhancing HSA convergence, this work proposes a new method for generating the initial harmonies (population of solutions) for NRP. In particular, we use a semi cyclic shift patterns in the initialization step to generate the initial harmonies (population) rather than using a fully random mechanism of basic HSA. The goal of this work is to investigate the effect of using different initialization method for HSA when solving NRP. To do this, we tested the proposed algorithm on a real world dataset obtained from a large hospital in Malaysia. Experimental results show that our proposed HSA produces better quality rosters for all considered instances than a basic HSA.

UNIVERSITI KEBANGSAAN MALAYSIA MEDICAL CENTRE (UKMMC)

Universiti Kebangsaan Malaysia Medical Center (UKMMC) is an educational hospital that belongs to National University Malaysia (UKM) working around the clock to provide its services for the public. The hospital has a workforce exceeding 1400 nurses attending more than 900 beds for 24 h a day and 7 days a week. For more details about the NRP of UKMMC refer to (Hadwan and Ayob, 2009, 2011; Hadwan *et al*., 2013).

Hard and soft constraints: Based on the abstraction of the information gathered from the interviews, observations and questionnaires at UKMMC, the problem we are dealing with has the following constraints:

UKMMC Hard constraints: Based on the abstraction of gathered data, we have the following hard constraints:

For the UKMMC NRP, we have identified the following hard constraints:

- **HC1:** The coverage demand for each shift must be fulfilled. Understaffing is not allowed
- **HC2:** All nurses work one shift per day at most
- **HC3:** One senior nurse must be allocated for every shift
- **HC4:** An isolated working day is prohibited. For instance, off-day followed by working-day then followed by off day
- **HC5:** For each fourteen days, the maximum working days are 12 days whilst the minimum are 10 days
- **HC6:** Each nurse works no more than 4 consecutive working days
- **HC7:** Night shift must be in the form of 4 consecutive night shifts followed by two days off

UKMMC soft constraints: Based on the analysis of the gathered data, there is a group of soft constraints which can be listed as follows:

- **SC1:** Attempts to allocate equal number of working days and days off to all nurses during the roster's period
- **SC2:** Attempts to allocate at least one day off in the weekend during the rostering period
- **SC3:** For the overall roster, try to allocate more morning and evening shifts than night shifts
- **SC4:** Attempts to allocate patterns of 4 consecutive morning shifts followed by one day off
- **SC5:** Attempts to allocate patterns of 4 consecutive evening shifts followed by one day off
- **SC6:** Attempt to give either a day off or evening shift after the two days off that follow night shift pattern (NNNNxxE) or (NNNNxxx)
- **SC7:** Attempts to give at least 12 h rest for the nurses before starting new shift. This would try not to give evening shift followed by morning (EM)

Based on the discussion with the head nurses at UKMMC, we found that (S1) and (S2) are more important than the others. Each soft constraint is associated with a particular weight. Normally, the higher weight indicates the main importance of satisfying the constraints. Refer to Table 1 for the weight of each soft constraint.

Coverage demand: The coverage demand is the most important constraint that must be satisfied. This constraint indicates the minimum required number of nurses and the needed skills combinations for each shift and day (Table 2). For example (Table 2), for CICU instance, the total number of nurses is 11 with 8 senior nurses and the coverage demand for weekdays

Table 1: The weight of soft constraints for each violation of the constraint

	Soft constraints	Penalty violation weight
W1	Gives a fair number of working days and days off to all staff nurses	100
W2	Gives each nurse at least one day off in the weekends	100
W3	Gives four consecutive mornings shift followed by one day off	10
W4	Gives four consecutive evenings shift followed by one day off	10
W5	Gives either a day off or an evening shift after the night shift pattern	1

Table 2: The minimum coverage demand for UKMMC dataset

			Weekdays (Mon-Fri)			Weekend (Sat-Sun)		
Instance/ward	No. of nurses	Senior nurses	Morning	Evening	Night	Morning	Evening	Night
CICU	11	8	3	3	2	2	2	2
SGY5	18	11	4	4	3	4	4	2
MD1	19	12	4	4	3	4	4	2
NICU	49	29	10	10	10	8	8	7
N50	50	31	10	10	10	10	10	10
ED	57	32	13	13	10	11	11	8
GICU	73	38	16	16	15	15	15	14
ICU	79	41	17	17	16	16	16	15

(morning, evening, night) and weekend (morning, evening, night) are (3, 3, 2) and (2, 2, 2), respectively. We deal with coverage demand as a hard constraint that cannot be violated at any time.

ENHANCED HARMONY SEARCH ALGORITHM FOR NRPS

Enhanced Harmony Search Algorithm (EHSA) is a HSA variant that employed the idea of semi-cyclic shift patterns approach that proposed by Hadwan and Ayob (2011) to construct the feasible shift patterns rather than filling up the harmony memory randomly. Also the idea of updating the HMCR and PAR dynamically is used rather than using fixed HMCR and PAR (Hasancebi *et al*., 2010). In the initialization step of basic HSA, the candidate solutions are filled up randomly to the HM (Geem *et al*., 2001; Lee and Geem, 2005; Geem, 2010). The procedure of basic HSA has five: Step 1. Initialize the HSA parameters and the optimized problem, Step 2. Build the Harmony Memory (HM), Step 3. Improvise new solution, Step 4. Update the HM, and Step 5. Repeat step 4 and 5 until reach one of the stopping criterion. Figure 1 presents the pseudo-code of building the HM with the modification of using a semi-cyclic shift pattern approach instead of the fully random mechanism of Basic HSA. However, the remaining steps were kept the same as basic HSA.

Figure 2 presents the pseudo-code of improvisation step where we used dynamic HMCR and PAR. In this modification, the parameter values of HMCR and PAR keep changed during the run based on the Eq. 1 and 2.

```
Building the Harmony Memory;
 Begin
  for (i=1 to HMS) do
     x_i=ø;
    while the coverage demand of night shift not met do
      allocate two patterns of one-week from the pool of solutions that include night shifts cyclically to x_i
    End while
    while not reaching the end of x_i do
      allocate two patterns of one-week from the pool of solutions of morning and evening shifts
    End while
   calculate the PV of x_i
   add x_i to HM
 end for
   sort the solutions based on its associated penalty values in HM.
End
```

Fig. 1: The pseudo-code of building the HM using SCSPA

```
Step 3 Improvise a new harmony (generate new solution)
begin
   x^new=ø
   for i=(1 to NI) do
            for j=(1 to n) do /* n is the number of nurses */
              for k=(1 to sp) do /* sp is the number of shift patterns */
             if (rand(0,1) =HMCR) then
                 choose p_i randomly from the HM:
                     if (rand(0,1)=PAR) then
                  pitch adjusted x^new [p_i] by:
                  x^new [p_i] = x^new [p_i]+ rand BW
                 else
                     x^new [p_i]=x[p_i]
                 endif
                 change the rate of PAR; /* According to equation (2) */
            else choose sp_i randomly from the pool of solutions
                x^new [p_i]= x[p_i]
            endif
            change the rate of HMCR; /* According to equation (1) */
           endfor
       endfor
   endfor
 end
```

Fig. 2: The pseudo code of improvisation step in EHSA

$$\mathrm{HMCR(gn)} = \mathrm{HMCR}_{min} + \frac{\mathrm{HMCR}_{max} - \mathrm{HMCR}_{min}}{\mathrm{NI}} \times g \quad (1)$$

$$\mathrm{PAR(gn)} = \mathrm{PAR}_{max} + \frac{\mathrm{PAR}_{max} - \mathrm{PAR}_{min}}{\mathrm{NI}} \times g \quad (2)$$

where, HMCR(gn) and PAR(gn) are the HMCR and PAR in generation g, respectively; NI is the maximum number of iterations and g is the current number of iterations; HMCR_{min} and HMCR_{max} are the minimum and the maximum of HMCR values respectively; PAR_{min} and PAR_{max} are the minimum and the maximum PAR values, respectively.

COMPUTATIONAL EXPERIMENTS AND RESULTS

Here, we report the results of applying the basic HSA (Geem *et al*., 2001) that implemented herein and the EHSA. The main goal of the following experiments is to measure the improvement of the convergence of the EHSA compared to the BHSA. The same experiments resources used to test the BHSA and EHSA. Table 3 shows the used parameter values of BHSA and EHSA. For BHSA, we experimentally identify the used parameter values in Table 3. For EHSA, the parameter values of HMCR_{min} and HMCR_{max} we followed Barzegari *et al*. (2010) and for PARmin and PARmax we followed the parameter values of Omran and Mahdavi (2008). For the HMS, BW and NI, we used the same parameter setting as in BHSA. We have used the UKMMC dataset to test EHSA and AHS.

Comparing the results of BHSA and EHSA: This section presents a comparison between the results obtained by applying the BHSA and the EHSA using the same problem instances and parameter values in Table 3. In Table 4, the average, the median, the Standard Deviation (Stddv.), the number of Desirable Patterns (DPs) and the execution time are used in the comparison. Based on the results presented in Table 4, the EHSA outperforms the BHSA in all the instances. EHSA produced better results compared with BHSA in terms of lower penalty values, number of Dps and execution time (highlighted in bold font in Table 4). For example, for CICU instance, the average penalty value obtained by EHSA is better than BHSA (i.e., 72.1 and 316.5, for EHSA and BHSA, respectively). This is attributed to the proposed changes to the way of building the initial HM and the use of the dynamic HMCR and PAR rather than using fixed values for these operators.

Figure 3 shows the box-and-whiskers plot for results (out of 20 runs) based on solution quality (i.e., penalty value, where the lower value is better) obtained by EHSA tested on UKMMC datasets. It shows that in most instances (five out of eight), most of the solutions (i.e., the median value) are closer (in term of quality of solution) to the best solutions.

Table 3: The parameter values used in BHSA and EHSA

	HMS	HMCR	PAR	BW	NI
BHSA	80	0.95	0.2	(5,-5)	50000
EHSA	80	HMCR_{min} 0.1 HMCR_{max} 0.95	PAR_{min} 0.01 PAR_{max} 0.99	(5,-5)	50000

Table 4: Comparison results (out of 20 runs): EHSA and BHSA on UKMMC dataset

	EHSA					BHSA				
Instance	Average	Median	SD	DPs	Time (sec)	Average	Median	SD	DPs	Time (sec)
CICU	72.1	81	2.781	10	35.1	341.1	316.5	40.781	5	55.1
SGY5	67	71	2.663	18	55.6	301.4	285	60.662	8	65.3
MD1	79.3	87	2.624	20	57.9	406.6	385.5	60.626	9	67
NICU	89	94.5	8.178	52	98	981.35	956.5	150.188	18	118.8
N50	121.9	128.5	10.784	55	105.4	1075.6	1044.5	180.689	21	125
ED	118	149.5	11.231	58	195.7	1194.4	1134	180.123	24	215.9
GICU	152.2	175.5	11.721	75	225	1607.1	1582.5	120.212	27	245.2
ICU	251.1	283.5	12.151	85	355.8	1773.1	1777	170.822	29	395.7

Table 5: Comparison results (out of 20 runs): EHSA and AHS on UKMMC dataset

	EHSA					AHS				
Instance	Average	Median	SD	DPs	Time (sec)	Average	Median	SD	DPs	Time (sec)
CICU	72.1	81	2.781	10	35.1	128.5	128.6	5.838	7	524.7
SGY5	67	71	2.663	18	55.6	159.8	158.2	10.018	9	571.6
MD1	79.3	87	2.624	20	57.9	172.2	172.4	8.702	10	587.6
NICU	89	94.5	8.178	52	98	304	303.5	11.487	21	617.1
N50	121.9	128.5	10.784	55	105.4	312.5	310.8	9.184	23	625.8
ED	118	149.5	11.231	58	195.7	431	431.5	8.182	25	674.8
GICU	152.2	175.5	11.721	75	225	671.3	670.7	9.926	29	723.3
ICU	251.1	283.5	12.151	85	355.8	731.3	729.5	12.852	33	795.7

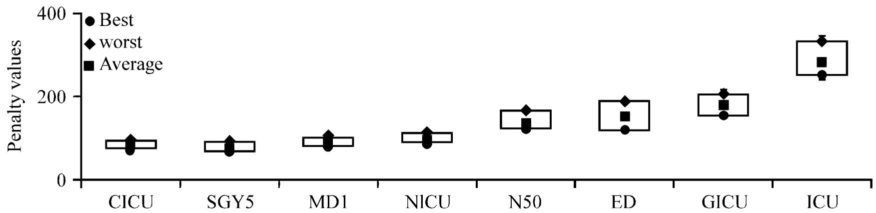

Fig. 3: Box-and-whiskers plot for the results (out of 20 runs) based on penalty values (y-axis) obtained by EHSA on UKMMC dataset (x-axis)

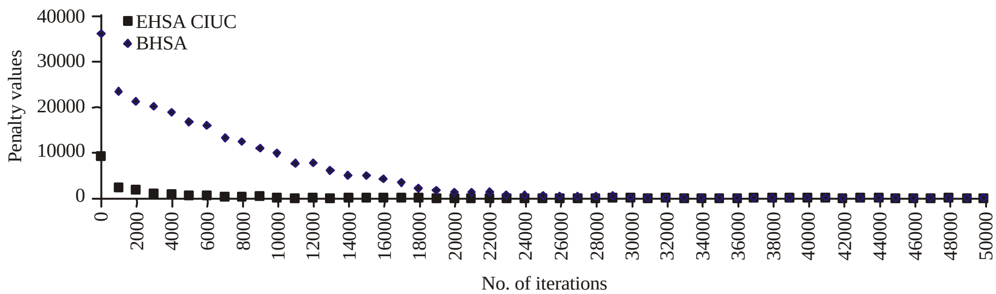

Fig. 4: The solution landscape (based on solution quality versus number of iteration) for CICU instance using BHSA and EHSA

```
The Improved Harmony Search Algorithm (AHS)
Step 2: Building the Harmony Memory (HM)
 begin
   for i = (1 to HMS) do
       xi=ø;
       for j=(1 to n) do /* n is the number of nurses */
               for k=(1 to sp) do /* sp is the number of shift patterns */
               choose variable pi randomly from the PS
               add pi to xi
            endfor
       endfor
       calculate the f(xi)
       add xi to HM
       sort the solutions based on its penalty values in HM
 endfor
 end
```

Fig. 5: The pseudo code of building the HM step in AHS

Figure 4 shows the solution landscape of the BHSA and the EHSA when solving CICU for instance. Where, the EHSA needs a lower number of iterations to develop the candidate solutions and could reach to lower penalty values after 10000 iterations. In contrast, the BHSA needs about 30000 iterations to reach closer to the penalty values obtained by the EHSA after being run for 10000 iterations. As the execution time is concerned, there is no remarkable difference between the BHSA and the EHSA. However, it is noticed that the EHSA was faster in all computational experiments.

A comparison between EHSA and AHS: Adaptive Harmony Search (AHS) is one of the new variants of the HSA proposed by Hasancebi *et al.* (2010). The main reason of comparing the EHSA with the AHS is that we used the same idea of updating the values of the HMCR and the PAR of both algorithms. However, the only point of difference is that the AHS uses a random mechanism in initializing the HM whilst we use the SCSPA mechanism in the EHSA. The same experiments design and parameter values as in Table 3 are used for both algorithms in order to make a fair comparison. Figure 5 and 6 present the

```
The Improved Harmony Search Algorithm (AHS)
Step 3 Improvise a new harmony (generate new solution)
begin
   x^new = ø
  for i=(1 to NI) do
            for j=(1 to n) do /* n is the number of nurses */
                for k=(1 to sp) do /* sp is the number of shift patterns */
                 if (rand(0,1) =HMCR) then
                    choose sp_i randomly from the HM:
                       if (rand(0,1)=PAR) then
                     pitch adjusted x^new [p_i] by:
                              x^new [p_i] = x^new [p_i]+ rand BW
                           else
                     x^new [p_i]=x[p_i]
                  endif
                        change the rate of PAR; /* According to equation (2) */
             else choose sp randomly from the pool of solutions
                     x^new[p_i]= x[p_i]
              endif
              change the rate of HMCR; /* According to equation (1) */
           endfor
       endfor
 endfor
end
```

Fig. 6: The pseudo code of improvisation step in AHS

pseudo code of building the HM and the improvisation of a new harmony, respectively.

Table 5 presents a comparison between the results of the EHSA and the AHS. Based on the results shown in Table 5, the EHSA outperforms the AHS in all the evaluation criteria. For example, for CICU instance, the average penalty value obtained by EHSA is better than AHS (i.e., 72.1 and 128.5, for EHSA and BHSA, respectively). This is attributed to the control of the randomness that we gain by using the SCSPA rather than the fully random manner that the AHS. Initializing the HM using the SCSPA helps in generating harmonies (population) with no violations of the constraints concerning the night shift patterns. Doing that, we have solved the most problematic part of the UKMMC dataset during the initialization step. This supports our assumption that using the SCSPA will reduce the violations of the night shift constraints and help the EHSA to start developing candidate solutions with a good quality (less penalty values).

CONCLUSION

This study has introduced the Enhanced Harmony Search Algorithm (EHSA) in which two modifications are employed in building the initial populations and controlling the main parameters of the improvisation step. We studied the behavior of the EHSA during the run in order to observe and compare its convergence with the BHSA. Then, we compared the EHSA and the BHSA in order to show the ability of the EHSA in overcoming the drawbacks of the BHSA in all the instances of UKMMC dataset. Another comparison that was conducted between the EHSA and the AHS shows that the EHSA outperform the AHS due to the poor initial populations that random mechanism generates using the AHS. In the future study, the attempt will be made to answer the following question "can the performance of the EHSA be further improved if we hybridize it with other local search algorithms?". It is highly recommended by many researchers such as Talbi (2009), Blum *et al.* (2011) and Qu *et al.* (2009) to investigate the hybridization of population based and local search-based algorithms.

ACKNOWLEDGMENTS

The authors wish to thank Ministry of Higher Education for supporting this work under the FRGS Research Grant Scheme (FRGS/1/2012/SG05/UKM/02/11).

REFERENCES

Aickelin, U. and K.A. Dowsland, 2004. An indirect genetic algorithm for a nurse-scheduling problem. Comput. Oper. Res., 31: 761-778.

Aickelin, U., E.K. Burke and J. Li, 2009. An evolutionary squeaky wheel optimization approach to personnel scheduling. IEEE Trans. Evol. Comput., 13: 433-443.

Al-Betar, M.A. and A.T. Khader, 2010. A harmony search algorithm for university course timetabling. Ann. Oper. Res., 194: 3-33.

Barzegari, M., S.M.T. Bathaee and M. Alizadeh, 2010. Optimal coordination of directional overcurrent relays using harmony search algorithm. Proceedings of the 9th International Conference on Environment and Electrical Engineering, May 16-19, 2010, Prague, Czech Republic, pp: 321-324.

Bilgin, B., P. De Causmaecker, B. Rossie and G. Vanden Berghe, 2012. Local search neighbourhoods for dealing with a novel nurse rostering model. Ann. Oper. Res., 194: 33-57.

Blum, C., J. Puchinger, G.R. Raidl and A. Roli, 2011. Hybrid metaheuristics in combinatorial optimization: A survey. Applied Soft Comput., 11: 4135-4151.

Blum, C., M.J.B. Aguilera, A. Roli and M. Samples, 2008. Hybrid Metaheuristics: An Emerging Approach to Optimization. Vol. 114, Springer, Heidelberg, Germany.

Burke, E.K., P. De Causmaecker, G.V. Berghe and H. Van Landeghem, 2004. The state of the art of nurse rostering. J. Scheduling, 7: 441-499.

Burke, E.K., T. Curtois, R. Qu and G.V. Berghe, 2010. A scatter search methodology for the nurse rostering problem. J. Oper. Res. Soc., 61: 1667-1679.

Cheang, B., H. Li, A. Lim and B. Rodrigues, 2003. Nurse rostering problems: A bibliographic survey. Eur. J. Oper. Res., 151: 447-460.

Coelho, L.D.S. and V.C. Mariani, 2009. An improved harmony search algorithm for power economic load dispatch. Energy Convers. Manage., 50: 2522-2526.

Degertekin, S.O., 2008. Optimum design of steel frames using harmony search algorithm. Struct. Multidi. Optim., 36: 393-401.

Geem, Z.W., J.H. Kim and G.V. Loganathan, 2001. A new heuristic optimization algorithm: Harmony search. Simulation, 76: 60-68.

Geem, Z.W., K.S. Lee and Y. Park, 2005. Application of harmony search to vehicle routing. Am. J. Applied Sci., 2: 1552-1557.

Geem, Z.W., 2010. Multiobjective optimization of time-cost trade-off using harmony search. J. Constr. Eng. Manage., 136: 711-716.

Glass, C.A. and R.A. Knight, 2009. The nurse rostering problem: A critical appraisal of the problem structure. Eur. J. Oper. Res., 202: 379-389.

Gutjahr, W.J. and M.S. Rauner, 2007. An ACO algorithm for a dynamic regional nurse-scheduling problem in Austria. J. Comput. Oper. Res., 34: 642-666.

Hadwan, M. and M. Ayob, 2011. A semi-cyclic shift patterns approach for nurse rostering problems. Proceedings of the 3rd Conference on Data Mining and Optimization, June 28-29, 2011, Putrajaya, Malaysia, pp: 184-189.

Hadwan, M. and M.B. Ayob, 2009. An exploration study of nurse rostering practice at hospital Universiti Kebangsaan Malaysia. Proceedings of the 2nd Conference on Data Mining and Optimization, October 27-28, 2009, Bangi, Selangor, Malaysia, pp: 100-107.

Hadwan, M., M. Ayob, N.R. Sabar and R. Qu, 2013. A harmony search algorithm for nurse rostering problems. Inform. Sci., 233: 126-140.

Hasancebi, O., F. Erdal and M.P. Saka, 2010. An adaptive harmony search method for structural optimization. J. Struct. Eng., 136: 419-431.

Ingram, G. and T. Zhang, 2009. Overview of Applications and Developments in the Harmony Search Algorithm. In: Music-Inspired Harmony Search Algorithm, Geem, Z.W. (Ed.). Springer, Berlin/Heidelberg, pp: 15-37.

Lee, K. and Z. Geem, 2005. A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Comput. Methods Applied Mechanics Eng., 194: 3902-3933.

Lee, K.S. and Z.W. Geem, 2004. A new structural optimization method based on the harmony search algorithm. Comput. Struct., 82: 781-798.

Mahdavi, M., M. Fesanghari and E. Damangir, 2007. An improved harmony search algorithm for solving optimization problems. Applied Math. Comput., 188: 1567-1579.

Omran, M.G. and M. Mahdavi, 2008. Global-best harmony search. Applied Math. Comput., 198: 643-656.

Qu, R., E.K. Burke, B. McCollum, L.T. Merlot and S.Y. Lee, 2009. A survey of search methodologies and automated system development for examination timetabling. J. Schedul., 12: 55-89.

Saka, M. and O. Hasancebi, 2009. Adaptive Harmony Search Algorithm for Design Code Optimization of Steel Structures. In: Harmony Search Algorithms for Structural Design Optimization, Geem, Z. (Ed.). Springer, Berlin, Germany, pp: 79-120.

Talbi, E.G., 2009. Metaheuristics: From Design to Implementation. John Wiley and Sons, USA., ISBN: 9780470496909, Pages: 500.

Thornton, J.R. and A. Sattar, 1997. Applied partial constraint satisfaction using weighted iterative repair. Proceedings of the 10th Australian Joint Conference on Artificial Intelligence: Advanced Topics in Artificial Intelligence, November 30-December 4, 1997, Australia, pp: 57-66.

Wang, C.M. and Y.F. Huang, 2010. Self-adaptive harmony search algorithm for optimization. Expert Syst. Appli., 37: 2826-2837.