Nurse Rostering Using Modified Harmony Search Algorithm

Mohammed A. Awadallah[1], Ahamad Tajudin Khader[1], Mohammed Azmi Al-Betar[1,2], and Asaju La'aro Bolaji[1]

[1] School of Computer Sciences, Universiti Sains Malaysia (USM), 11800, Pulau Pinang, Malaysia
[2] Department of Computer Science, Al-zaytoonah University, Amman, Jordan
{mama10_com018,abl10_sa0739}@student.usm.my, {tajudin,mohbetar}@cs.usm.my

Abstract. In this paper, a Harmony Search Algorithm (HSA) is adapted for Nurse Rostering Problem (NRP). HSA is a global optimization method derived from a musical improvisation process which has been successfully tailored for several optimization domains. NRP is a hard combinatorial scheduling problem of assigning given shifts to given nurses. Using a dataset established by International Nurse Rostering Competition 2010 of sprint dataset that has 10-early, 10-late, 10-hidden, and 3-hint. The proposed method achieved competitively comparable results.

1 Introduction

The Nurse Rostering Problem (NRP) is normally tackled by assigning a set of shifts of different types to a limited number of nurses with different working skills and working contracts. Four main factors should be considered during the construction of a nurse roster: hospital management policies, government regulations, fair shift distribution among nurses, and nurses' preferences [1]. Basically, these factors are classified into hard and soft constraints based on the hospital administrative perspective. Hard constraints are those that should be satisfied, while violations of the soft constraints are allowed but should be avoided, if possible. From hospital administrative perspectives, the difficulties of constructing the nurse roster arise due to a considerable number of constraints which vary amongst hospitals. A nurse roster can be said to be *feasible* by satisfying hard constraints but its quality is determined by satisfying soft constraints. However, it is almost impossible to find a roster that fulfills all constraints. Computationally, this is a combinatorial optimization problem, which belongs to an NP-complete class in almost all its variations [2]. NRP is a hard scheduling problem which does not lend itself to be solved by manual methods.

In the past, the head nurse used to generate a nurse roster based on accumulated personal experience by taking into account the set of constraints to enhance the quality of the nurse roster. This would consume considerable time without even meeting most constraints. Hence, operational researchers and artificial intelligence

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part II, LNCS 7077, pp. 27–37, 2011.

experts have been directing their attention to solve NRP. Different approaches have therefore been proposed, such as Tabu Search [3], Variable Neighbourhood Search [4][5], Simulated Annealing [6], Ant Colony Optimization [7], Genetic Algorithm [8], Electromagnetic Algorithm [9], Scatter Search [10], Memetic Algorithm [11]. More information about these and other methods can be seen in the surveys [1][12].

Harmony Search Algorithm (HSA) is a new population-based metaheuristic proposed by Geem et al., [13]. It has been successfully applied to a wide variety of optimization problems such as structural design [14], vehicle routing [15], water network design [16], tour routing [17], traveling salesman problem [13], course timetabling [18][19], examination timetabling [20], and many others as reported in [21][22]. HSA has different characteristics: (i) it is easy to tailor for different types of optimization problems; (ii) it requires few mathematical requirements and does not require a derived value of the decision variables in the initial stage of search [23]. Therefore, the performance is improved by tuning HS parameters [24][25], hybrid with other methods such as particle swarm optimization [26][27][28], and ant colony algorithm [29].

HSA is derived from the behavioral phenomenon of musicians in the improvisation process, where a set of musicians play pitches of their instruments repetitively to come up with pleasing harmony as determined by an aesthetic standard value. Analogously in optimization, a set of decision variables is assigned with values, iteratively to come up with a (near) optimal solution as determined by the objective function. Practically, the HSA is an iterative process that starts with a set of initial solutions stored in harmony memory (HM). At each iteration, a new solution (new harmony) is generated and evaluated to replace the worst solution in the HM, if it is better. This process is repeated until a stop criterion is met.

The main objective of this paper is to alert HSA for NRP as an initial exploration to investigate the effectiveness of such method in the nurse rostering domain, henceforth called Modified Harmony Search Algorithm (MHSA).

For evaluation purposes, a dataset established by the organizers of International Nurse Rostering Competition 2010 (INRC2010) is used. Note that this is the first standard dataset for NRP with its instances divided into three categories: sprint, middle, and long distance. These are different in terms of size and complexity. Each category is further classified into four types: early, hidden, late, and hint. More details can been found in INRC2010 website[1]. In a nutshell, MHSA has been able to yield comparable results.

The outline of the paper is as follows: In Sect. 2, we describe the nurse rostering problem we were dealing with. In Sect. 3 and 4 we discuss the algorithm and results respectively. In Section 5 we draw conclusions on the success of this approach and present some possible future extensions.

2 Problem Description

NRP consists of a set of nurses to be assigned to a set of different shifts on daily basis over given time periods. Each nurse has a specific title with skills (i.e., Head Nurse,

[1] http://www.kuleuvenkortrijk.be/nrpcompetition

Nurse) determined by qualification and experience. Furthermore, each nurse is employed through a contract agreed upon with the hospital administration (i.e., Full time, Half time). These contracts that determine the nurse job specifications as formulated in the soft constraints ($S_1, \ldots, S_6, S_9, S_{10}$) are shown in Table 1.

Table 1. Description of INRC2010 Hard (H_1, H_2) and Soft ($S_1, \ldots, S_{10}$) constraints

Symbol	The constraint
H_1	All demanded shifts must be assigned to a nurse.
H_2	A nurse can only work one shift per day, i.e. no two shifts can be assigned to the same nurse on a day.
S_1	Maximum and minimum number of assignments for each nurse during the scheduling period.
S_2	Maximum and minimum number of consecutive working days.
S_3	Maximum and minimum number of consecutive free days.
S_4	Assign complete weekends.
S_5	Assign identical complete weekends.
S_6	Two free days after a night shift.
S_7	Requested day-on/off.
S_8	Requested shift-on/off.
S_9	Alternative skill.
S_{10}	Unwanted patterns. (Where pattern is a set of legal shifts defined in terms of work to be done during the shifts [30]).

Table 2 gives detailed explanations for each shift which consists of the skill of the nurse, start and end time, and the number of nurses required for each day (i.e., shift demand).

Table 2. Shift Categories Details

Shift Details					Shift Demand						
Shift		Skill	Start Time	End Time	Mon	Tue	Wed	Thu	Fri	Sat	Sun
D	Day Shift	Nurse	08:30	16:30	3	3	3	3	3	2	2
L	Late Shift	Nurse	14:30	22:30	9	9	9	9	9	5	5
E	Early Shift	Nurse	06:30	14:30	9	9	9	9	9	5	5
N	Night Shift	Nurse	22:30	06:30	3	3	3	3	3	2	2
DH	Head Nurse Day Shift	Head Nurse	08:30	16:30	2	2	2	2	2	1	1

It is worth mentioning that the nurse roster should satisfy the nurse preferences (see S_7, S_8 in Table 1), for example the nurse preferences Day-OFF/ON (i.e., the nurse prefers (not) to work on a specific day) or Shift-OFF/ON (i.e., the nurse prefers (not) to be assigned to a specific shift on a specific day). Data for such preferences are gathered from the nurses well before handling the scheduling process.

Conventionally, the constraints in NRP are divided into two types: hard and soft constraints. Hard constraints (H_1, H_2) are those that should be satisfied, while the fulfillment of the soft constraints (S_1 ,… , S_{10}) is desired but not absolutely essential. The basic objective is to find a roster that satisfies all hard constraints while minimizing the penalty of soft constraint violations. The mathematical formulation of the two hard constraints is as follows:

H_1: All demanded shifts must be assigned to a nurse (see (1)).

$$\sum_{i=1}^{N} x_i = d_{jk}. \tag{1}$$

H_2: A nurse can only work one shift per day (see (2)).

$$\sum_{i=1}^{N} x_i \leq 1. \tag{2}$$

Where x_i is allocation in nurse roster solution ($\boldsymbol{x}$) assigned with a triple of items (nurse u, day v, shift r). d_{jk} is the number of nurses required for day (j) at shift (k). Note that, $v = j$, $r = k$, and N is the maximum length of allocations for solution roster ($\boldsymbol{x}$) calculated as in (3).

$$N = \sum_{i=0}^{W-1} \sum_{j=1}^{7} \sum_{k=1}^{T} d_{((i\times 7)+j)k}. \tag{3}$$

In (3). (W) is the maximum number of weeks in a scheduling period, while (T) is the total number of shifts.

The nurse roster is evaluated using an objective function (see (4)) that adds up the penalty of soft constraint violations in a feasible roster.

$$min \;\; f(\boldsymbol{x}) = \sum_{s=1}^{10} c_s . g_s(x). \tag{4}$$

Note that s refers to the index of the soft constraint (S_1, …, S_{10}), c_s refers to the penalty weight for the violation of the soft constraint s, and $g_s(\boldsymbol{x})$ is the total number of violations in $\boldsymbol{x}$ for the soft constraint s, where $\boldsymbol{x}$ is a roster solution as formulated in Fig. 1.

3 Modified Harmony Search Algorithm for NRP

The HSA is an optimization technique inspired by the music improvisation process. Naturally, musicians play pitches of their instruments relying on both experiences and randomness realized in their off-hand skills. In the optimization process, decision variables can be assigned based on accumulative search or randomness.

The HSA is a population-based method starting with a set of vectors stored in *Harmony Memory* (HM). At each iteration, a new vector (i.e., *new harmony*) is generated based on three operators: (i) *Memory Consideration*, which makes use of accumulative search from HM vectors (i.e., same functionality as the crossover operator in Genetic Algorithm (GA)); (ii) *Random Consideration*, which is used to diversify the new harmony (i.e., same functionality as the mutation operator in GA), and (iii) *Pitch Adjustment*, which is responsible for the local improvement (i.e., same functionality as move in local search). The objective function is used to evaluate the quality of the new harmony. Iteratively, if the new harmony has a better quality than the worst vector in HM, the HSA will substitute the worst vector with the new harmony and this process is repeated until a stopping criterion is met.

This section thoroughly describes the methodology followed in this paper with detailed explanation of how to modify HSA steps into NRP.

STEP 1. Initialize the parameters of the NRP and HSA. Within the NRP, the parameters are normally drawn from the dataset instance to be processed. These parameters include the set of nurses, the set of skill categories, the set of shift types, the scheduling period times, the set of work contracts, the set of cover demand requests, the set of preferences of nurses, and eventually the set of unwanted patterns. Each contract contains the details of agreement between the hospital and the nurse that include: maximum/minimum number of assignment shifts, maximum/minimum number of consecutive working days, maximum/minimum number of consecutive free days, maximum/minimum number of consecutive working weekends, maximum number of working weekends within four weeks, the days of weekend, and unwanted patterns (see Table 1).

The objective function described in (4) is utilized to evaluate each roster generated by HSA. Fig. 1 displays the roster $\boldsymbol{x}$ which includes a set of allocations each of which takes a value of a combination of nurses, days, and shifts. The possible range for each allocation is within all possible combinations of nurses, days, and shifts. The parameters of the HSA required to solve the optimization problem are also specified in this step: (i) The Harmony Memory Consideration Rate (HMCR), determines the rate of selecting the values from HM vectors. (ii) The Harmony Memory Size (HMS), determines the number of initial vectors in HM. (iii) The Pitch Adjustment Rate (PAR), determines the rate of the local improvement. (iv) The Number of Improvisations (NI) corresponds to the number of iterations that are required to solve NRP which are also defined.

x_1	x_2	x_3	x_4	. . .	x_{N-1}	x_N
Nurse 1 Day 1 Shift D	Nurse 9 Day 12 Shift L	Nurse 2 Day 3 Shift N	Nurse 2 Day 16 Shift E	. . . .	Nurse 4 Day 10 Shift D	Nurse 10 Day 28 Shift DH

Fig. 1. Roster $\boldsymbol{x}$ representation

STEP 2. Initialize the harmony memory. The Harmony Memory (**HM**) consists of a set of all feasible rosters as determined by HMS (see (5)). In this step, these rosters are constructed using a heuristic ordering method [31] whereby the shifts will be sorted in ascending order based on their difficulty level, and then the required nurses of the ordered shifts will be assigned first. These rosters are sorted in ascending order based on their objective function values.

$$\mathrm{HM} = \left[\begin{array}{cccc|c} x_1^{\;1} & x_2^{\;1} & \dots & x_N^{\;1} & f(\boldsymbol{x}^1) \\ x_1^{\;2} & x_2^{\;2} & \dots & x_N^{\;2} & f(\boldsymbol{x}^2) \\ \dots & \dots & \dots & \dots & \dots \\ x_1^{\;HMS} & x_2^{\;HMS} & \dots & x_N^{\;HMS} & f(\boldsymbol{x}^{HMS}) \end{array}\right] \qquad (5)$$

STEP 3. Improvise a new harmony roster. In this step, a new harmony roster, $\boldsymbol{x}' = (x_1', x_2', \dots, x_N')$, is constructed from scratch based on three operators: (i) memory consideration, (ii) random consideration, and (iii) pitch adjustment. The new roster must be complete and feasible. It should be emphasized that in certain iteration, if the HSA cannot construct a complete and feasible roster, the repair procedure will be triggered.

Memory Consideration operator. The memory consideration operator selects a feasible value for allocation x_i' in the new harmony roster from the values of the same allocations stored in HM rosters such that $x_i' \in \{x_i^1, x_i^2, \dots, x_i^{HMS}\}$, for $\forall i \in (1, 2, ..., N)$ with probability (w.p.) of HMCR where HMCR $\in [0,1]$. Notice that the feasibility must be always maintained.

Random Consideration operator. The allocation x_i' that met the probability of (1-HMCR) will be assigned by a random value from its possible range where the rules of heuristic ordering method have been considered. The process of Memory Consideration and Random Consideration can be summarized as follows:

$$x_i' = \begin{cases} x_i' \in \{x_i^1, x_i^2, \dots, x_i^{HMS}\} & \text{w.p.} \quad \text{HMCR}, \\ x_i' \in \boldsymbol{X}_i' & \text{w.p.} \quad (1 - \text{HMCR}). \end{cases}$$

Where X_i' is a set of all feasible values for allocation x_i'.

Pitch Adjustment operator. The allocation x_i' assigned by memory consideration will be pitch adjusted with probability of PAR where PAR $\in [0,1]$ as follows:

$$\text{Pitch adjustment for } x_i' = \begin{cases} \mathit{Yes} & \text{w.p.} \quad \text{PAR}, \\ \mathit{No} & \text{w.p.} \quad (1 - \text{PAR}). \end{cases}$$

For NRP, different nighbourhood structures have been used to improve the roster locally. Here, the pitch adjustment operator is divided into four local changes: (i) ***Move***, (ii) ***Swap1***, (iii) ***Swap2***, and (iv) ***Switch***. Each of which is controlled by a specific PAR range as follows:

$$x_i' = \begin{cases} \textit{Move} & 0 \le U(0,1) < (PAR/4), \\ \textit{Swap1} & (PAR/4) \le U(0,1) < (2\times PAR/4), \\ \textit{Swap2} & (2\times PAR/4) \le U(0,1) < (3\times PAR/4), \\ \textit{Switch} & (3\times PAR/4) \le U(0,1) < PAR, \\ \textit{Do nothing} & PAR \le U(0,1) \le 1. \end{cases}$$

The four proposed pitch adjustment procedures are designed to run as follows:

1. ***Move:*** with probability of [0, PAR/4), the nurse of the selected allocation x_i' will be changed to another nurse randomly to solve the violation of the soft constraint S4 (see Fig.2. (a)).
2. ***Swap1:*** with probability of [PAR/4, 2×PAR/4), the shift of the selected allocation x_i' will be exchanged with another shift on the same day for another selected allocation x_j' to solve the violation of the soft constraint S_{10} (see Fig.2. (b)).
3. ***Swap2:*** with probability of [2×PAR/4, 3×PAR/4), the shift of the selected allocation x_i' will be exchanged with another shift on the same day for another selected allocation x_j' to solve or minimize the violations of the soft constraints S_5, S_8, S_{10} (see Fig.2. (c)).
4. ***Switch:*** with probability of [3×PAR/4, PAR), the shift of the selected allocation x_i' will be exchanged with another shift with the same nurse for another selected allocation x_j' to solve or minimize the violations of the soft constraints S_5, S_8, S_{10} (see Fig.2.(d)).

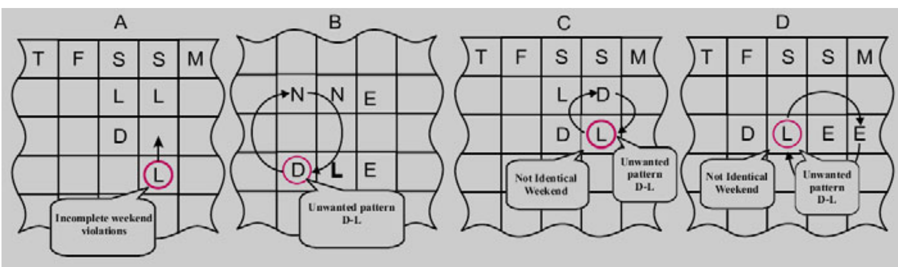

Fig. 2. Pitch Adjustment procedures

In this paper, any local changes that do not improve the new harmony or result in an unfeasible roster will be discarded. Note that the other constraints not covered in the pitch adjustment procedures; do not apply to a partial roster, but rather to a complete one.

STEP 4. Repair the new harmony. When the improvisation (STEP 3) process of the new harmony roster is completed, it must be checked for feasibility. However, if the feasibility is not achieved, then the repair process is triggered to render the new harmony roster feasible using the following two steps: Firstly, identify the allocations which are not scheduled in the new harmony roster. Secondly, use a random selection to find feasible values for unscheduled allocations in the new harmony roster. Yet, if the repair process with the predefined iterations is not successful in solving the feasibility issue, then discard the new harmony roster and restart the improvisation process from scratch.

STEP 5. Update the harmony memory. If the new roster, $\boldsymbol{x}' = (x_1', x_2', \ldots, x_N')$, is better than the worst roster in **HM**, the new roster replaces the worst roster.

STEP 6. Check the stop criterion. Based on NI step 3, step 4, and Step 5 of HSA are repeated.

4 Experiments and Results

The MHSA evaluated in this section is programmed using Microsoft Visual C++ 6.0 under WinXP on an Intel Machine with CoreTM processor 2.66GHz, and 2GB RAM. A dataset provided by INRC2010 for nurse rostering is used. The Sprint group of INRC2010 dataset used here consists of 33 problem instances (10 sprint early, 10 sprint late, 10 sprint hidden, 3 sprint hint). The characteristics of the dataset are overviewed in Table 3. The conducted experiments are designed to investigate the effectiveness of MHSA for NRP. Each dataset ran 10 times. The parameter settings are: HMS =100, HMCR=0.99, PAR=0.001 and NI is between 100000 and 300000.

Table 3. Characteristics of the sprint group of INRC2010 dataset

Characteristic	Early	Late	Hidden	Hint
Number of nurses	10	10	10	10
Number of skills	1	1	1	1
Number of shifts	4	4	3, 4	4
Number of contracts	4	3	3	3
Number of patterns	3	0, 3, 4, 8	4, 8	0, 8
Period of schedule	1 to 28 Jan. 2010	1 to 28 Jan. 2010	1 to 28 Jun. 2010	1 to 28 Jan. 2010
Day-Off request	√	√, or *X*	√	√
Shift-Off request	√	√, or *X*	√	√

Table 4 shows the results produced by MSHA in terms of best, average, worst and standard deviation. Moreover, the competition results are recorded for the purpose of comparison. Note that the numbers in the table refer to the penalty values of the violations on the soft constraints (lowest is best). The best results are highlighted in bold while the underlined results are the close-to-the-best results achieved by MHSA. The values of differences between the best cited result and MHSA results are also shown in the *Diff.* column. The symbol ' √ ' indicates that the used method obtains the best result while the symbol ' - ' denotes that it cannot obtain the best result.

Apparently, the results are comparable and impressive, yet cannot precisely measure up to the best cited results. However, these results can be excused as this method is experimented with to initially explore its efficiency for NRP.

The results are compared with competition participants" methods abbreviated as follows[2]

- G1: A hyper-heuristic combined with a greedy shuffle approach by Burak Bilgin, Peter Demeester, Mustafa Misir, Wim Vancroonenburg, Greet Vanden Berghe, and Tony Wauters.

[2] `http://www.kuleuven-kortrijk.be/.u0041139/nrpcompetition/abstracts/`

- G2: An ejection chain method and a branch and price algorithm by Edmund K. Burke, and Tim Curtois.
- G3: Adaptive Local Search, by Zhipeng Lu and Jin-Kao Hao.
- G4: General Constraint Optimization Solver by Koji Nonobe.
- G5: A systematic two-phase approach by Christos Valouxis, Christos Gogos, George Goulas, Panayiotis Alefragis and Efthymios Housos.

Table 4. The results achieved by MHSA for NRP

	MHSA Results				Best		Competition Winners				
Instance Name	Best	Worst	Average	Std. div.	Results	Diff.	G1	G2	G3	G4	G5
Sprint Early 01	60	73	65.1	3.59	56	4	-	√	√	√	√
Sprint Early 02	61	76	66.4	4.22	58	3	-	√	√	√	√
Sprint Early 03	56	65	59.9	2.98	51	5	-	√	√	√	√
Sprint Early 04	66	81	71.9	4.46	59	7	√	√	-	√	√
Sprint Early 05	61	69	65	2.61	58	3	√	√	√	√	√
Sprint Early 06	58	69	63.6	3.29	54	4	√	√	√	√	√
Sprint Early 07	62	81	65.8	5.38	56	6	√	√	√	√	√
Sprint Early 08	59	65	62.8	2.04	56	3	-	√	√	√	√
Sprint Early 09	57	69	63.6	3.47	55	2	-	√	√	√	√
Sprint Early 10	58	70	62	3.52	52	6	-	√	√	√	√
Sprint Late 01	47	58	53.8	3.43	37	10	-	√	-	-	√
Sprint Late 02	53	65	58.1	3.08	42	11	-	√	-	-	√
Sprint Late 03	59	71	65.1	4.04	48	11	-	√	-	√	√
Sprint Late 04	117	138	127.6	6.33	75	42	-	√	-	-	-
Sprint Late 05	54	63	57.5	3.07	44	10	-	√	-	-	√
Sprint Late 06	47	68	54.6	6.39	42	5	-	√	√	√	√
Sprint Late 07	66	107	87.3	11.40	42	24	-	√	-	-	-
Sprint Late 08	19	81	45.1	17.28	17	2	-	√	√	√	√
Sprint Late 09	34	99	55.5	18.33	17	17	-	√	√	√	√
Sprint Late 10	73	116	95.7	13.39	43	30	-	√	-	-	-
Sprint Hidden 01	48	74	54.8	6.76	33	15	-	-	-	√	√
Sprint Hidden 02	45	61	52.1	4.64	32	13	√	-	-	√	-
Sprint Hidden 03	76	89	81.4	3.61	62	14	-	-	-	√	√
Sprint Hidden 04	97	214	178.4	33.99	67	30	-	-	-	√	√
Sprint Hidden 05	68	89	77.5	7.10	59	9	√	-	-	-	-
Sprint Hidden 06	278	977	481.9	199.68	134	144	-	-	-	√	-
Sprint Hidden 07	201	374	292.4	48.71	153	48	-	-	-	-	√
Sprint Hidden 08	374	488	432	31.94	209	165	-	-	-	√	-
Sprint Hidden 09	916	1357	1232.8	125.50	338	578	-	-	-	-	√
Sprint Hidden 10	462	605	542	42.13	306	156	-	-	-	-	√
Sprint Hint 01	104	133	118.9	8.68	78	26	-	√	-	-	-
Sprint Hint 02	73	98	87.7	7.25	47	26	-	√	-	-	-
Sprint Hint 03	92	131	112.8	10.68	57	35	-	√	-	-	-

5 Conclusion and Future Work

This paper has presented a Modified Harmony Search Algorithm (MHSA) for the Nurse Rostering Problem (NRP) as an initial investigation to experiment with the method for tackling NRP. Firstly, heuristic ordering has been used to generate feasible solutions that satisfy all hard constraints. Secondly, harmony search operators are adapted to realize the problem domain knowledge. Note that the feasible search space region is dealt with. MHSA is able to generate a new roster in each iteration;

the roster is globally improved using the memory consideration, random consideration, and is locally improved using pitch adjustment. Using the International Nurse Rostering Competition 2010 (INRC2010) dataset, the MHSA is able to produce a feasible roster with competitively comparable results.

In the future, a sensitivity analysis for the HSA parameters for NRP has to be carried out, and more work could be done on developing new neighbourhood techniques based on different problem constraints or hybridizing MHSA with local search-based methods. We believe that powerful local changes will substantially improve the quality of the results.

Acknowledgments. The third author is grateful to be awarded a Postdoctoral Fellowship from the school of Computer Sciences (USM).

References

1. Aickelin, U., Dowsland, K.: An indirect genetic algorithm for a nurse-scheduling problem. Computers & Operations Research 31(5), 76–778 (2004)
2. Al-Betar, M., Khader, A.: A hybrid harmony search for university course timetabling. In: Proceedings of the 4nd Multidisciplinary Conference on Scheduling: Theory and Applications (MISTA 2009), Dublin, Ireland, pp. 10–12 (August 2009)
3. Al-Betar, M., Khader, A.: A harmony search algorithm for university course timetabling. Annals of Operations Research, 1–29 (2008)
4. Al-Betar, M., Khader, A., Nadi, F.: Selection mechanisms in memory consideration for examination timetabling with harmony search. In: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, pp. 1203–1210. ACM (2010)
5. Alatas, B.: Chaotic harmony search algorithms. Applied Mathematics and Computation 216(9), 2687–2699 (2010)
6. Bilgin, B., De Causmaecker, P., Rossie, B., Vanden Berghe, G.: Local search neighbourhoods for dealing with a novel nurse rostering model. Annals of Operations Research, 1–25 (2011)
7. Brusco, M., Jacobs, L.: Cost analysis of alternative formulations for personnel scheduling in continuously operating organizations. European Journal of Operational Research 86(2), 249–261 (1995)
8. Burke, E., Cowling, P., De Causmaecker, P., Berghe, G.: A memetic approach to the nurse rostering problem. Applied Intelligence 15(3), 199–214 (2001)
9. Burke, E., Curtois, T., Post, G., Qu, R., Veltman, B.: A hybrid heuristic ordering and variable neighbourhood search for the nurse rostering problem. European Journal of Operational Research 188(2), 330–341 (2008)
10. Burke, E., Curtois, T., Qu, R., Berghe, G.: A scatter search methodology for the nurse rostering problem. Journal of the Operational Research Society 61(11), 1667–1679 (2009)
11. Burke, E., De Causmaecker, P., Berghe, G., Van Landeghem, H.: The state of the art of nurse rostering. Journal of Scheduling 7(6), 441–499 (2004)
12. Burke, E., De Causmaecker, P., Vanden Berghe, G.: A hybrid tabu search algorithm for the nurse rostering problem. Simulated Evolution and Learning, 187–194 (1999)
13. Burke, E., Li, J., Qu, R.: A hybrid model of integer programming and variable neighbourhood search for highly-constrained nurse rostering problems. European Journal of Operational Research 203(2), 484–493 (2010)

14. Cheang, B., Li, H., Lim, A., Rodrigues, B.: Nurse rostering problems - a bibliographic survey. European Journal of Operational Research 151(3), 447–460 (2003)
15. Geem, Z.: Optimal cost design of water distribution networks using harmony search. Engineering Optimization 38(3), 259–277 (2006)
16. Geem, Z.: Harmony search applications in industry. Soft Computing Applications in Industry, 117–134 (2008)
17. Geem, Z., Kim, J., Loganathan, G.: A new heuristic optimization algorithm: harmony search. Simulation 76(2), 60–68 (2001)
18. Geem, Z., Lee, K., Park, Y.: Application of harmony search to vehicle routing. American Journal of Applied Sciences 2(12), 1552–1557 (2005)
19. Geem, Z., Sim, K.: Parameter-setting-free harmony search algorithm. Applied Mathematics and Computation, 3881–3889 (2010)
20. Geem, Z., Tseng, C., Park, Y.: Harmony search for generalized orienteering problem: best touring in china. Advances in Natural Computation, 741–750 (2005)
21. Gutjahr, W., Rauner, M.: An ACO algorithm for a dynamic regional nurse-scheduling problem in Austria. Computers & Operations Research 34(3), 642–666 (2007)
22. Kaveh, A., Talatahari, S.: Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures. Computers & Structures 87(5-6), 267–283 (2009)
23. Lee, K., Geem, Z.: A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Computer Methods in Applied Mechanics and Engineering 194(36-38), 3902–3933 (2005)
24. Lee, K., Geem, Z., Lee, S., Bae, K.: The harmony search heuristic algorithm for discrete structural optimization. Engineering Optimization 37(7), 663–684 (2005)
25. Maenhout, B., Vanhoucke, M.: An electromagnetic meta-heuristic for the nurse scheduling problem. Journal of Heuristics 13(4), 359–385 (2007)
26. Mahdavi, M., Fesanghary, M., Damangir, E.: An improved harmony search algorithm for solving optimization problems. Applied Mathematics and Computation 188(2), 1567–1579 (2007)
27. Millar, H., Kiragu, M.: Cyclic and non-cyclic scheduling of 12 h shift nurses by network programming. European Journal of Operational Research 104(3), 582–592 (1998)
28. Pan, Q., Suganthan, P., Liang, J., Tasgetiren, M.: A local-best harmony search algorithm with dynamic subpopulations. Engineering Optimization 42(2), 101–117 (2010)
29. Pan, Q., Suganthan, P., Tasgetiren, M., Liang, J.: A self-adaptive global best harmony search algorithm for continuous optimization problems. Applied Mathematics and Computation 216(3), 830–848 (2010)
30. Wren, A.: Scheduling, Timetabling and Rosteringa Special Relationship? In: Burke, E.K., Ross, P. (eds.) PATAT 1995. LNCS, vol. 1153, pp. 46–75. Springer, Heidelberg (1996)
31. Zhao, S., Suganthan, P., Pan, Q., Tasgetiren, M.: Dynamic multi-swarm particle swarm optimizer with harmony search. Expert Systems with Applications, 3735–3742 (2010)