Hybrid Harmony Search Algorithm for Nurse Rostering Problem

Yabing Nie, Bing Wang and Xianxia Zhang

Abstract This paper addresses the nurse rostering problem (NRP), whose objective is to minimize a total penalty caused by the roster. A large number of constraints required to be considered could cause a great difficulty of handling the NRP. A hybrid harmony search algorithm (HHSA) with a greedy local search is proposed to solve the NRP. A personal schedule is divided into several blocks, in which a subset of constraints is considered in advance. Based on these blocks, the pitch adjustment and randomization are carried out. Every time a roster is improvised, a coverage repairing procedure is applied to make the shift constraints satisfied, and the greedy local search is used to improve the roster's quality. The proposed HHAS was tested on many well known real-world problem instances and competitive solutions were obtained.

Keywords Harmony search algorithm · Greedy local search · Nurse rostering problem

1 Introduction

The nurse rostering problem (NRP) is a complex combinatorial problem whose objective is to produce rosters which satisfy all hard constraints while taking into account soft constraints. Hard constraints are those constraints that must be satisfied at all costs for the NRP. If some requirements are desirable but could be violated, these requirements are usually referred to as soft constraints for the NRP. The solution quality of NRP could be evaluated by estimating soft constraints.

A wide variety of approaches have been developed to solve the NRP, including exact algorithms, meta-heuristic algorithms and others [1]. Due to the computational

Y. Nie · B. Wang(✉) · X. Zhang
School of Mechatronic Engineering and Automation, Shanghai University,
Shanghai 200072, China
e-mail: susanbwang@shu.edu.cn

J.H. Kim and Z.W. Geem (eds.), *Harmony Search Algorithm*,
Advances in Intelligent Systems and Computing 382,
DOI: 10.1007/978-3-662-47926-1_12

complexity of large scale NRPs, meta-heuristic algorithms were more popular than exact algorithms. Various meta-heuristic algorithms were developed to solve the NRP, including Genetic Algorithm [2], Scatter Search [3], Tabu Search [4], Simulated Annealing [5] and many others. The hybridizations of these algorithms have also attracted the interest of the researches. Post and Veltman [6] proposed a hybrid genetic algorithm, in which a local search was carried out after each generation of the genetic algorithm to make improvement. Burke et al. [7] developed a hybrid variable neighborhood search (VNS) which created an initial schedule by an adaptive ordering technique in advance and then run variable VNS based on this initial schedule. Burke et al. [8] developed an IP-based VNS which used an IP to first solve a small problem including the full set of hard constraints and a subset of the soft constraints. All the remaining constraints were then satisfied by a basic VNS.

The NRP is commonly described and solved by three views: a nurse-day view, a nurse-task view and a nurse-shift pattern view [9]. The nurse-day view and nurse-task view are similar. Accordingly, the NRP is solved by generating assignments of nurses to shift for each day of the planning period, such as [3, 6, 7, 8]. However, in the nurse-shift pattern view, rosters are built by constructing personal schedules for each individual nurse (by allocating one of his/her feasible shift patterns to him/her), such as [2, 4, 5].

HSA is a new population-based algorithm mimicking the improvisation process of searching for a perfect state of harmony measured by aesthetic standards. Since developed by Geem et al. [10], it was successfully used in various optimization problems such as university timetabling [11], vehicle routing [12] and so on. HSA was also used to solve the NRP, which was usually solved by the nurse-task view, such as [13, 14]. However, Hadwan et al. [15] adapted a harmony search algorithm to solve the NRP by the nurse-shift pattern view.

In this work, rosters were also produced by the nurse-shift pattern view, but with different pitch adjustment operator and different way of handling the coverage constraint from [15]. Moreover, a greedy local search was carried out after each generation of the HSA to make improvement, which was similar to [6].

The remainder of this paper is organized as follows. In section 2, the nurse rostering problem is described. In Section 3, we first briefly present the classic HSA and then detail our HHSA. Experiment was conducted to investigate the algorithm developed in Section 4. In section 5, we draw conclusions on the success of this HHSA.

2 Problem Description

Here, we use an ORTEC NRP as a case to study. The problem was fully described in [7], except for several specific individual rostering constraints which were not explicitly listed in the academic papers but were embedded in the solutions published on the NRP benchmark web site [16]. The length of planning period is one month (31 days), from Wednesday 1st January to Friday 31st January 2003. Four shift types (i.e. early, day, late and night) are assigned to 16 nurses. The nurses have three different contracts: standard 36, standard 32 and standard 20. For completeness, all constraints that need to be satisfied are presented briefly in the following.

The problem has the following hard constraints:

(HC1) The number of people on each shift of each day must exactly be the specified level. Both overstaffing and understaffing are not allowed.

(HC2) A nurse can work at most one shift per day.

(HC3) A nurse can work at most 3 night shifts per period of 5 consecutive weeks.

(HC4) A nurse can work at most 3 weekends per 5 week period.

(HC5) After a series of consecutive night shifts at least two days off are required.

(HC6) The number of consecutive night shifts is at most 3.

(HC7) No isolated night shift is allowed in a personal schedule.

(HC8) Each nurse has his/her maximum number of working days.

(HC9) The number of consecutive shifts is at most 6.

(HC10) No late shifts for one particular nurse.

(HC11) There is minimum time limit between shifts.

In addition, the problem has the following soft constraints:

(SC1) Either two days on duty or two days off at weekends for each nurse

(SC2) There is no night shift before a free weekend.

(SC3) For any nurse avoid stand-alone shifts.

(SC4) The number of consecutive night shifts has minimum and maximum limits.

(SC5) After a series of working days at least two free days are required.

(SC6) The number of working days per week has minimum and maximum limits.

(SC7) The number of consecutive working days has minimum and maximum limits.

(SC8) The number of consecutive early shifts has minimum and maximum limits.

(SC9) The number of consecutive late shifts has minimum and maximum limits.

(SC10) Certain shift type successions (e.g. day shift followed by early shift) should be avoided.

Hard constraints are constraints that must be satisfied at all costs so as to obtain a feasible solution. Soft constraints can be violated if necessary, but at the cost of incurring penalty. The objective of the NRP is to minimize the total penalty caused by the roster. The solution of this problem is a roster which is made up of the personal schedules of all nurses. Each personal schedule is represented as a string. The j[th] position in the string of nurse i is represented as follows.

$$p_{ij} = \begin{cases} "O" & nurse\ i\ works\ no\ shift\ on\ day\ j \\ "E" & nurse\ i\ works\ on\ early\ shift\ on\ day\ j \\ "D" & nurse\ i\ works\ on\ day\ shift\ on\ day\ j \\ "L" & nurse\ i\ works\ on\ late\ shift\ on\ day\ j \\ "N" & nurse\ i\ works\ on\ night\ shift\ on\ day\ j \end{cases} \tag{1}$$

The difficulty of the NRP is due to the large number and a variety of constraints that need to be satisfied. When the nurse-shift pattern view is used to solve the NRP, constraints are usually further divided into two categories to be handled: shift constraints and nurse constraints [17]. Shift constraints specify the number of nurses required for each shift during the entire planning period, such as (HC1). Nurse constraints refer to all the restrictions on personal schedules including personal requests, personal preferences, and constraints on balancing the workload among personnel, such as (HC2) ~ (HC11) and (SC1) ~ (SC10). Because all soft constraints in our case belong to nurse constraints, thus the penalty of each personal schedule can be calculated in isolation from other personal schedules. Adding the penalties of all personal schedules up the penalty of an overall roster is obtained.

3 HHSA for the Nurse Rostering Problem

3.1 Harmony Search Algorithm

HSA is a population-based meta-heuristic algorithm, developed in an analogy with musical improvisation. In music performance, each music player improvises one note at a time. All these musical notes are combined together to form a harmony, evaluated by aesthetic standards and improved through practice after practice. In optimization, each variable is assigned a value at a time. All these values are combined together to form a solution vector, evaluated by the objective function and improved iteration by iteration. The main steps in the structure of HSA are as follows:

Step 1: Initialize the algorithm parameters.
Step 2: Initialize the harmony memory (HM).
Step 3: Improvise a new solution from the HM.
Step 4: Update the HM.
Step 5: Repeat step 3 and step 4 until the stopping criterion is satisfied.

3.1.1 Initialization of Algorithm Parameters

The parameters of HSA are specified in this step, including harmony memory size (HMS), harmony memory consideration rate (HMCR), pitch adjustment rate (PAR) and the maximum number of improvisations (NI). Both HMCR and PAR are used to improvise a new solution in step3.

3.1.2 Initialization of HM

In this step, the HM is initially stuffed with as many randomly generated solutions as the HMS. Those solutions are sorted by the values of objective function. The structure of the HM is shown as following:

$$HM = \begin{bmatrix} x^1 \\ x^2 \\ \vdots \\ x^{HMS-1} \\ x^{HMS} \end{bmatrix} = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_{N-1}^1 & x_N^1 \\ x_1^2 & x_2^2 & \cdots & x_{N-1}^2 & x_N^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^{HMS-1} & x_2^{HMS-1} & \cdots & x_{N-1}^{HMS-1} & x_N^{HMS-1} \\ x_1^{HMS} & x_2^{HMS} & \cdots & x_{N-1}^{HMS} & x_N^{HMS} \end{bmatrix} \quad (2)$$

3.1.3 Improvising a New Solution From the HM

In this step, a new solution is improvised based on the following three rules: memory consideration, pitch adjustment and randomization. In the improvising procedure, as shown in Fig. 1, each variable of the new solution is assigned a value in turn.

```
For each i ∈ [1, N] do
    If U(0,1) ≤ HMCR then
        Assign a value to x_i using memory consideration
        If U(0,1) ≤ PAR then
        Pitch adjust the value obtained by memory consideration
        End if
    Else
        Assign a value to x_i using randomization
    End if
End for
```

Fig. 1 The Pseudo-code of Improvising a New Solution

3.1.4 Updating the HM

If the newly made solution is better than the worst one in the HM, measured in terms of the objective function value, include the new solution in the HM and exclude the existing worst solution in the HM. Otherwise, discard the new solution.

3.1.5 Checking the Stopping Criterion

Repeat step3 and step4 until the stopping criterion is satisfied. Normally the stopping criterion defines the maximum number of improvisations (NI).

3.2 The HHSA for the Nurse Rostering Problem

A key factor in the application of HSA is how the algorithm handles the constraints relating to the NRP. In the improvising procedure of proposed HHSA, each nurse is allocated a personal schedule at a time while considering nurse constraints and avoiding overstaffing. All these personal schedules are combined together to form a roster. Every time a roster is built an additional repair process is employed to assign enough people to understaffed shifts and then a greedy local search is applied to improve the roster's quality. In the following, we mainly elaborate the process of improvising new rosters in Section 3.2.1 and Section 3.2.2. The repair process and the greedy local search are presented in Section 3.2.3 and Section 3.2.4 respectively.

3.2.1 The Construction of Weekly Shift Patterns

In the improvising procedure of traditional HSA, we randomly move the variable value obtained by the memory consideration to a neighboring value via pitch adjustment with probability PAR or randomly select one variable value from possible range of values via randomization with probability (1-HMCR). However, each personal schedule, i.e. variable value, is highly constrained by nurse constraints in the NRP. Thus the personal schedules generated by general pitch adjustment or randomization are likely to violate many nurse constraints, which can cause to poor performance of the algorithm and considerable increase in iterations need to find an optimal solution. In order to improve the efficiency of HSA, we divide a personal schedule into several blocks, in which a subset of constraints is considered. Based on these blocks the pitch adjustment and randomization are carried out.

One block represents one week, in which a weekly shift pattern will be put. Fig. 2 gives an example of the relationship between blocks and a complete personal schedule. It is worth mentioning that a working week runs from Monday to Sunday in the problem. Thus lengths of the first week and last week in the January are 5 due to the structure of the month itself.

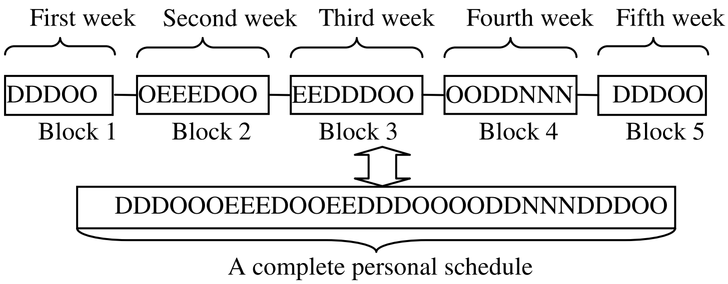

Fig. 2 The Relationship between Blocks and a Complete Personal Schedule

For each nurse, his/her valid weekly shift patterns for each week are generated as follows:

1) Generating all valid 2-day and 3-day shift sequences by systematically generating all the possible permutations of all shift types (E, D, L, N and O) over 2 days and 3 days.
2) For each individual nurse, combining the 2-day and 3-day shift sequences to form all his/her valid weekly shift patterns corresponding to each week by avoiding violating hard constraints and highly weighted soft constraints.

The reasons why we divide a complete personal schedule into several blocks by the week are following: 1) The length of a week is neither too long nor too short and thus is easy to tackle; 2) A number of constraints in our case are to restrict the shift content of a week, such as (SC1), (SC2), (SC6) and others. The violations of these constraints can be avoided during the construction of weekly shift patterns as far as possible or at least be determined to evaluate the qualities of generated weekly shift patterns. By only combining the weekly shift patterns with low penalties (≤ 50), the complete personal schedule formed is likely to have better quality and thus the solution space can be reduced effectively.

It is worth mentioning that the qualities of weekly shift patterns are just optimistic estimates, by only concerning the determined constraint violations. To elaborate, assuming that there is a weekly shift pattern 'ODOEEOO' for the second week of the month, in which there are two single 'O'. We can determine that the second 'O' violates (SC5), but we can not determine whether the first 'O' violates (SC5), because we do not know what the shifts are in the first week. In the same way, we check every soft constraint and calculate the penalty occurred by each weekly shift pattern.

3.2.2 The Improvisation of New Rosters Based on Weekly Shift Patterns

Each roster in the initial HM is generated randomly. After repaired and improved, as described in section 3.2.3 and section 3.2.4, these rosters are included and sorted in the initial HM. Then new rosters are improvised from the HM.

As described in section 2, both understaffing and overstaffing are not allowed. In the algorithm, the overstaffing is avoided during the improvising procedure and the understaffing is avoided by a coverage repairing procedure after the roster is improvised.

(1) Memory consideration
For each nurse to be scheduled, one of his/her possible personal schedules in the HM is randomly selected as his/her personal schedule in the new roster via memory consideration with probability HMCR. The shifts causing overstaffing in the selected personal schedule are set to be "O".

(2) Pitch adjustment
The personal schedule obtained by memory consideration will be pitch adjusted with probability of PAR. As described in section 3.2.1, a complete personal schedule is divided into several blocks by the week. In the pitch adjustment operator, one of these blocks will be assigned another weekly shift pattern to improve the quality of the selected personal schedule. Which block will be changed is controlled by a specific PAR range as follows:

$$\textit{the pitch adjustment} \begin{cases} \textit{change block } 1 & 0 \le U(0,1) < (PAR/5) \\ \textit{change block } 2 & (PAR/5) \le U(0,1) < (2*PAR/5) \\ \textit{change block } 3 & (2*PAR/5) \le U(0,1) < (3*PAR/5) \\ \textit{change block } 4 & (3*PAR/5) \le U(0,1) < (4*PAR/5) \\ \textit{change block } 5 & (4*PAR/5) \le U(0,1) < PAR \end{cases} \quad (3)$$

After deciding which block to change, each valid weekly shift pattern causing no overstaffing will be put into this block and combined with other fixed blocks to form a neighbor personal schedule. The neighbor personal schedule with lowest penalty will be selected at last. However, if there is no weekly shift pattern could be used to form a neighbor personal schedule, add the personal schedule obtained by memory consideration to the new roster without changing.

It is worth mentioning that in order to evaluate the qualities of personal schedules if they are infeasible, the hard constraints (HC3) ~ (HC11) are also attached with very large weights in the HHSA, from 1000 to 10000. However, we emphasize that there is no violation of hard constraints allowed in the final roster.

(3) Randomization
If a personal schedule is constructed by the randomization operator, each block of the personal schedule will be randomly assigned a valid weekly shift pattern. The shifts causing overstaffing in the personal schedule are set to be "O".

Based on the three rules, personal schedules for each nurse are constructed in turn. All these personal schedules are combined together to form a new roster.

3.2.3 The Coverage Repairing Procedure

Though overstaffing has been avoided in the improvising procedure, there may still be some understaffed shifts in the new roster improvised. Thus a repair process is triggered to eliminate all the understaffed shifts by a greedy heuristic, in which each understaffed shift is added to the nurse's personal schedule whose penalty decreases the most (or increases the least if all worsen) on receiving this shift until there is no understaffing [18].

3.2.4 The Greedy Local Search

After this repair step, an efficient greedy local search is carried out on the roster to improve its quality. This greedy local search has been embedded in many methods [5, 18]. It simply swaps any pair of consecutive shifts between two nurses in the roster as long as the swaps decrease the penalty of the roster. To avoid violating shift constraints again, swaps will only be made vertically. To elaborate, Fig. 3 is used to show all the possible swaps between nurse 1 and nurse 3 within a 3-day period. The greedy search stops until no further improvement can be made. Then the improved roster is used to update the HM.

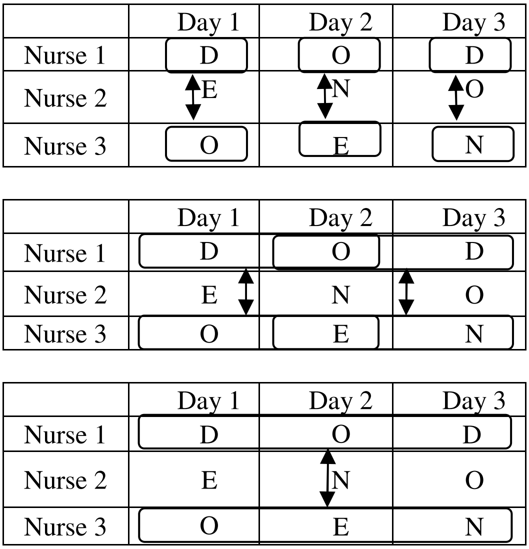

Fig. 3 All the Possible Swaps between Nurse 1 and Nurse 3 within a 3-day Period

4 Computational Results

The proposed HHSA were tested on a real-world problem with twelve data instances, which was first described by [5]. Each instance represented a month. It is worth mentioning that these instances were designed only to produce roster for an isolated month on the assumption that previous roster was empty. The HHSA parameter values were set as following: $HMS = 5$, $HMCR = 0.95$, $PAR = 0.2$, $NI = 1000$ .The experiments were performed on a PC with a Intel(R) Core(TM) i7-4790 CPU 3.60GHz processor and Windows 7 operating system. We run each of the instances 10 times. The results of the HHSA are showed in Table 1. Each result was obtained within 0.5 hours.

Table 1 Solutions obtained by the HHSA

Data	Best	Average	Worst
Jan	431	475	530
Feb	1510	1543	1575
Mar	3555	3641	3760
Apr	261	343	445
May	1900	2237	2935
Jun	10000	10187	10195
Jul	255	331	460
Aug	4381	4471	4575
Sept	271	330	465
Oct	396	479	540
Nov	1571	1620	1675
Dec	230	268	300

Table 2 Comparison of the HHSA with existing algorithms

Data	Hybrid GA [6]	Hybrid VNS [7]	Hybrid IP [8]	HHSA
Jan	775	735	460	**431**
Feb	1791	1866	1526	**1510**
Mar	2030	2010	1713	3555
Apr	612	457	391	**261**
May	2296	2161	2090	**1900**
Jun	9466	9291	8826	10000
Jul	781	481	425	**255**
Aug	4850	4880	3488	4381
Sept	615	647	330	**271**
Oct	736	665	445	**396**
Nov	2126	2030	1613	**1571**
Dec	625	520	405	**230**

Results in Table 2 demonstrate that the HHSA is able to obtain competitive results for 9 out of 12 instances compared with existing hybrid meta-heuristic algorithms. It is interesting to observe that although the constraints for each month are quite similar, the penalties for each month are quite different. These differences are caused by the structure of the month itself. For example, in the instance June, 1st June is a Sunday. Given that previous roster is empty, the nurses working on 1st June will automatically gain an unavoidable penalty by not working a complete weekend.

5 Conclusions

In this paper, a hybrid algorithm HHSA is proposed to solve the nurse rostering problem. The function of HSA is to globally and locally improvise new rosters and a greedy local search is used to improve the qualities of these rosters. In the improvising procedure of HHSA, both the pitch adjustment and randomization are carried out on high quality weekly shift patterns that are constructed in advance. By doing this, the personal schedules formed are likely to close to optimal schedule, and thus the roster consist of them can provide a good backbone, based on which the greedy local search can start from promising areas in the search space. The proposed hybrid algorithm were tested on a real-world problem with twelve data instances and obtained competitive results in a reasonable time.

Acknowledgement This work is partly supported by National Natural Science Foundation of China #61273182.

References

1. Burke, E.K., Causmaecker, P.De, Berghe, G.V., Landeghem, H.V.: The state of the art of nurse rostering. J. Sched. **7**, 441–499 (2004)
2. Aickelin, U., Dowsland, K.A.: An indirect genetic algorithm for a nurse scheduling problem. Comput. Oper. Res. **31**, 761–778 (2004)
3. Burke, E.K., Curtois, T., Qu, R., Berghe, G.V.: A scatter search methodology for the nurse rostering problem. J. Oper. Res. Soc. **61**, 1667–1679 (2010)
4. Dowsland, K.A.: Nurse scheduling with tabu search and strategic oscillation. Eur. J. Oper. Res. **106**, 393–407 (1998)
5. Burke, E.K., Li, J., Qu, R.: A pareto-based search methodology for multi-objective nurse scheduling. Ann. Oper. Res. **196**, 91–109 (2012)
6. Post, G., Veltman, B.: Harmonious personnel scheduling. In: Proceedings of 5th International Conference on Practice and Automated Timetabling, pp. 557–559 (2004)
7. Burke, E.K., Curtois, T., Post, G., Qu, R.: A hybrid heuristic ordering and variable neighbourhood search for the nurse rostering problem. Eur. J. Oper. Res. **188**, 330–341 (2008)
8. Burke, E.K., Li, J., Qu, R.: A hybrid model of integer programming and variable neighbourhood search for highly-constrained nurse rostering problems. Eur. J. Oper. Res. **203**, 484–493 (2010)

9. Cheang, B., Li, H., Lim, A., Rodrigues, B.: Nurse rostering problems—a bibliographic survey. Eur. J. Oper. Res. **151**, 447–460 (2003)
10. Geem, Z.W., Kim, J.H., Loganathan, G.V.: A new heuristic optimization algorithm: harmony search. Simulation **76**, 60–68 (2001)
11. Al-Betar, M.A., Khader, A.T.: A harmony search algorithm for university course timetabling. Ann. Oper. Res. **194**, 3–31 (2012)
12. Geem, Z.W., Lee, K.S., Park, Y.: Application of harmony search to vehicle routing. Am. J. Appl. Sci. **2**, 1552–1557 (2005)
13. Awadallah, M.A., Khader, A.T., Al-Betar, M.A., Bolaji, A.L.: Nurse rostering using modified harmony search algorithm. Swarm, Evolutionary, and Memetic Computing **7077**, 27–37 (2011)
14. Awadallah, M.A., Khader, A.T., Al-Betar, M.A., Bolaji, A.L.: Nurse scheduling using harmony search. In: 6th International Conference on Bio-Inspired Computing: Theories and Applications, pp. 58–63 (2011)
15. Hadwan, M., Ayob, M., Sabar, N.R., Qu, R.: A harmony search algorithm for nurse rostering problems. Inform. Sciences. **233**, 126–140 (2013)
16. Curtois, T.: Personnel scheduling data sets and benchmarks. University of Nottingham. http://www.cs.nott.ac.uk/~tec/NRP/
17. Ikegami, A., Niwa, A.: A subproblem-centric model and approach to the nurse scheduling problem. Math. Program. **97**, 517–541 (2003)
18. Brucker, P., Burke, E.K., Curtois, T., Qu, R., Berghe, G.V.: A shift sequence based approach for nurse scheduling and a new benchmark dataset. J. Heuristics **16**, 559–573 (2010)