Journal of King Saud University – Computer and Information Sciences (2013) **25**, 145–162

King Saud University

**Journal of King Saud University –
Computer and Information Sciences**

www.ksu.edu.sa
www.sciencedirect.com

ORIGINAL ARTICLE

Global best Harmony Search with a new pitch adjustment designed for Nurse Rostering

Mohammed A. Awadallah [a,*], **Ahamad Tajudin Khader** [a],
Mohammed Azmi Al-Betar [a,b], **Asaju La'aro Bolaji** [a]

[a] *School of Computer Sciences, Universiti Sains Malaysia, 11800 Pulau Pinang, Malaysia*
[b] *Department of Computer Science, Jadara University, Irbid, Jordan*

Received 25 May 2012; revised 5 September 2012; accepted 8 October 2012
Available online 16 October 2012

KEYWORDS

Nurse Rostering;
Harmony Search;
Approximation method;
Population-based;
Global-best

Abstract In this paper, the Harmony Search Algorithm (HSA) is proposed to tackle the Nurse Rostering Problem (NRP) using a dataset introduced in the First International Nurse Rostering Competition (INRC2010). NRP is a combinatorial optimization problem that is tackled by assigning a set of nurses with different skills and contracts to different types of shifts, over a predefined scheduling period. HSA is an approximation method which mimics the improvisation process that has been successfully applied for a wide range of optimization problems. It improvises the new harmony iteratively using three operators: memory consideration, random consideration, and pitch adjustment. Recently, HSA has been used for NRP, with promising results. This paper has made two major improvements to HSA for NRP: (i) replacing random selection with the Global-best selection of Particle Swarm Optimization in memory consideration operator to improve convergence speed. (ii) Establishing multi-pitch adjustment procedures to improve local exploitation. The result obtained by HSA is comparable with those produced by the five INRC2010 winners' methods.

1. Introduction

Nurse Rostering Problem (NRP) is tackled by assigning qualified nurses to a set of different shifts over a predefined scheduling period. Solving NRP is subject to two types of constraints: *hard* and *soft*. The hard constraints must be fulfilled to obtain *feasible* roster while the violations of soft constraints are allowed but should be avoided, if possible. The *quality* of the roster is evaluated based on the fulfillments of the soft constraints. Based on the above, the basic objective is to obtain a feasible roster with high quality. However, it is almost impossible to find a roster that satisfies all constraints, since this problem is classified as a combinatorial optimization problem (Bartholdi, 1981; Millar and Kiragu, 1998).

Over the past years, there have been many methods proposed by researchers from the fields of operations research and artificial intelligence to tackle NRP. Such methods have

---

[*] Corresponding author.
E-mail addresses: mama10_com018@student.usm.my (M.A. Awadallah), tajudin@cs.usm.my (A.T. Khader), mohbetar@cs.usm.my (M.A. Al-Betar), abl10_sa0739@student.usm.my (A.L. Bolaji).

Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

1319-1578 © 2013 Production and hosting by Elsevier B.V. on behalf of King Saud University.
http://dx.doi.org/10.1016/j.jksuci.2012.10.004

been classified into two classes: *exact* and *approximation*. Exact methods are used to obtain an exact solution, which includes integer and liner programming (Maenhout and Vanhoucke, 2010; Millar and Kiragu, 1998). Nowadays, the exact methods have been used to find a partial solution for NRP, and the rest portion is completed by approximation methods (Burke et al., 2010). In contrast, approximation methods seek to obtain (near-) optimal solutions with a reasonable computational time. These methods are classified into two types: *local search-based* and *population-based* (Blum and Roli, 2003). Local search-based methods consider one solution from the search space at a time, which iteratively changes to reach local optima. Several local search-based methods have been investigated for tackling NRP such as Tabu Search (Burke et al., 1999; Dowsland, 1998), Variable Neighborhood Search (Bilgin et al., 2011; Burke et al., 2008), and Simulated Annealing (Brusco and Jacobs, 1995). Population-based methods consider a population of solutions from the search space at a time; these solutions are iteratively recombined and changed to find a global optimum. Several population-based methods are introduced for tackling NRP such as Genetic Algorithm (Aickelin and Dowsland, 2004; Tsai and Li, 2009), Ant Colony Optimization (Gutjahr and Rauner, 2007), Electromagnetic Algorithm (Maenhout and Vanhoucke, 2007), Scatter Search (Burke et al., 2009). More details about some of these methods can be seen in the surveys by Cheang et al. (2003) and Burke et al. (2004).

In this paper, we investigate the NRP introduced by the First International Nurse Rostering Competition (INRC2010). INRC2010 was organized by the CODeS research group at Katholieke Universiteit Leuven in Belgium, SINTEF Group in Norway and the University of Udine in Italy. The dataset presented by INRC2010 was classified into three tracks: sprint, medium, and long datasets which are different in complexity and size. Each track is categorized into four types in accordance with the publication time in the competition: early, late, hidden, and hint. For this challenge, there are several methods proposed to solve the INRC2010 dataset.

Valouxis et al. (2010) used Integer Programming (IP) to compete in INRC2010. The solution method consists of two phases: the first includes assigning different nurses to working days while the second schedules the nurses assigned to working days and certain shifts. For medium and long track datasets, the authors used three additional neighborhood structures in the first phase: (i) rescheduling one day in the roster for another time, (ii) rescheduling two days in the roster for another time, and (iii) reshuffling the shifts among nurses. This method ranked first in all three tacks. Burke and Curtois (2010) used two methods to tackle the INRC2010 dataset. The ejection chain-based method is used for the sprint track dataset while the branch and price method is used for medium and long track datasets. These methods ranked second for medium and long tracks, and secured the fourth rank in sprint track. Nonobe (2010) modeled the problem as Constraint Optimization Problem (COP), and then used the "COP solver" based on tabu search to compete in INRC2010. This technique came second, third, fourth in sprint, medium and long tracks, respectively.

Lu and Hao (2010) applied tabu search to tackle the competition dataset. The solution method had two phases: (i) a random heuristic method was used to get a feasible roster, and (ii) the two neighborhood structures (i.e., move and swap) were used to improve the solution. The method kept the previous rosters in an "elite pool". If the local search procedure cannot improve the quality of the roster within a given number of iterations, one of the elite rosters is randomly selected and the method restarts the second phase. Lu and Hao (2010) approach ranked third and fourth in the sprint and medium tracks, respectively. Bilgin et al. (2010) hybridized a hyper-heuristic with a greedy shuffle move to compete in INRC2010. The simulated annealing hyper-heuristic was initially used to generate a feasible roster and tried to satisfy the soft constraints as much as possible. The greedy shuffle was used in the improvement loop. Bilgin et al. hybrid method came third in long track, and fifth in sprint and medium tracks. Rizzato et al. (2010) used a heuristic method for solving the INRC2010 dataset. The heuristic method constructed a feasible roster while simultaneously trying to satisfy five pre-defined soft constraints. Furthermore, three local search procedures were used after constructing the roster for more enhancements. This method achieved the fifth position in long track. It is worth noting that no exact solution has as yet been found for the INRC2010 dataset and, therefore, there is more room for investigation. For the purpose of our study, the Harmony Search Algorithm is investigated for NRP using the INRC2010 dataset.

The Harmony Search Algorithm (HSA) is an approximation method proposed by Geem et al. (2001). It has been successfully applied to a wide variety of optimization problems such as the blocking permutation flow shop scheduling problem (Wang et al., 2010), the optimal power flow problem (Sivasubramani and Swarup, 2011), the multicast routing problem (Forsati et al., 2008), water distribution networks (Geem, 2006), course timetabling (Al-betar and Khader, 2009; Al-betar et al., 2012b), examination timetabling (Al-betar et al., 2010b; Al-betar et al., 2010c), protein structure prediction problem (Abual-rub et al., 2012), and many others reported in (Alia and Mandava, 2011; Ingram and Zhang, 2009). HSA has attracted the attention of several researchers to experiment with it due to its impressive characteristics: (i) it has novel derivative criteria (Geem, 2008), (ii) it requires fewer mathematical derivation in the initial search, and (iii) it iteratively generates a new solution by considering all existing solutions in the population (Lee and Geem, 2005).

HSA mimics the musical improvisation process in which a group of musicians play the pitches of their musical instruments together, seeking a pleasing harmony as determined by audio-aesthetic standards. It is considered a population-based algorithm with local search-based concepts (Lee et al., 2005). HSA starts with a population of solutions. It improvises the new harmony iteratively using three operators: memory consideration that selects the variables of the new harmony from harmony memory solutions, random consideration that is used for randomness to diversify the new harmony, and pitch adjustment that is used to improve the new harmony locally. In each iteration, a new harmony is generated, which substitutes the worst solution in the harmony memory. This process is repeated until it converges.

To overcome some of the raised shortcomings in the memory consideration and pitch adjustment operators, Mahdavi et al. (2007) proposed adaptive PAR and bw values to empower the exploitation capability of the pitch adjustment operator. Furthermore, Omran and Mahdavi (2008) used the Global-best idea of particle swarm optimization (PSO) for the pitch adjustment operator to improve the convergence. The survival of the fittest principle of the natural phenomenon is integrated with the memory consideration operator by means of substituting the random selection of the memory consideration with Global-best, proportional, tournament, linear ranking, and exponential ranking selection schemes to improve the selection capability of this operator (Al-betar et al. (2012a)). The Global-best of memory consideration operator is used in this paper.

Recently, Al-betar et al. (2010a) used HSA with multi-pitch adjustment procedures to solve course timetabling problems with impressive results. Other studies (Awadallah et al., 2011a; Awadallah et al., 2011b) proposed HSA to tackle NRP using INRC2010 dataset obtaining promising results. In this paper, two improvements are provided to HSA for NRP: (i) the Global-best selection of Particle Swarm Optimization replace the random selection in the improvisation process to increase the speed of convergence, and (ii) multi-pitch adjustment procedures are established to improve the exploitation capability. The proposed method is evaluated against the INRC2010 dataset, where HSA is able to produce impressive results.

This paper is organized as follows: Section 2 discusses the Nurse Rostering Problem, while the Harmony Search Algorithm for Nurse Rostering Problem is described in Section 3. Section 4 discusses the experimental results and compares them with the best results of the winners' methods reported on the INRC2010 website.[1] A conclusion and possible research directions are provided in Section 5.

2. Nurse Rostering Problem

The Nurse Rostering Problem (NRP) is tackled by assigning a set of nurses with various skills and contracts to a set of shift types over a scheduling period. The NRP solution (or roster) is subject to hard and soft constraints. The hard constraints (see below H_1, H_2) must be fulfilled in the roster. The fulfillment of soft constraints (see below $S_1 - S_{15}$) is desirable, and determines the quality of the roster. The basic objective is to find a roster that satisfies all hard constraints while minimizing soft constraint violations.

The NRP consists of a set of m nurses, $\boldsymbol{N} = \{n_0, n_1, \ldots, n_{m-1}\}$, each has a specific skill from the set of skill categories $\boldsymbol{K} = \{k_0, k_1, \ldots, k_{q-1}\}$, where q is the total number of skill categories. Each nurse has a specific contract from the set of contracts $\boldsymbol{C} = \{c_0, c_1, \ldots, c_{w-1}\}$, where w is the total number of contracts. Each day during the scheduling period $\boldsymbol{D} = \{d_0, d_1, \ldots, d_{b-1}\}$, is split into r different shift types, $\boldsymbol{S} = \{s_0, s_1, \ldots, s_{r-1}\}$. The total number of time slots is $p = (b \times r)$, where $\boldsymbol{T} = \{t_0, t_1, \ldots, t_{p-1}\}$ is the set of time slots. A nurse will be assigned to different shifts over the scheduling period restricted by the number of nurses required (i.e., demand requirement) $dmnd_{j,k}$ for each shift s_k in each day d_j. Also, the unwanted patterns $\boldsymbol{PAT} = \{pat_0, pat_1, \ldots, pat_{u-1}\}$ are determined, where u is the total number of patterns.

Table 1 contains the notation used to formalize the INRC2010 datasets, while the mathematical formulation for the constraints is provided below.

H_1: All demanded shifts must be assigned to a nurse is as follows:

$$\sum_{j=0}^{(b-1)} \sum_{k=0}^{(r-1)} \sum_{i=0}^{(m-1)} x_{i,(j \times r+k)} = dmnd_{j,k}. \tag{1}$$

H_2: A nurse can only work one shift per day is as follows:

$$\sum_{i \in N} \sum_{j \in D} \sum_{k=0}^{(r-1)} x_{i,(j \times r+k)} \leqslant 1. \tag{2}$$

S_1: Maximum number of assignments for each nurse during the scheduling period is as follows: $\forall i \in N$, and $\forall f \in C$

$$g_1(\boldsymbol{x}) = \max\left(\left(\sum_{j=0}^{(b-1)} \sum_{k=0}^{(r-1)} x_{i,(j \times r+k)} - maxSh_{i,f}\right), 0\right). \tag{3}$$

S_2: Minimum number of assignments for each nurse during the scheduling period is as follows: $\forall i \in N$, and $\forall f \in C$

$$g_2(\boldsymbol{x}) = \max\left(\left(minSh_{i,f} - \sum_{j=0}^{(b-1)} \sum_{k=0}^{(r-1)} x_{i,(j \times r+k)}\right), 0\right). \tag{4}$$

S_3: Maximum number of consecutive working days is as follows: $\forall i \in N$, and $\forall f \in C$

$$g_3(\boldsymbol{x}) = \sum_{z=0}^{(b-maxWD_{i,f}-1)} \max\left(\left(\sum_{j=z}^{(z+maxWD_{i,f})} \sum_{k=0}^{(r-1)} x_{i,(j \times r+k)} - maxWD_{i,f}\right), 0\right). \tag{5}$$

S_4: Minimum number of consecutive working days is as follows: $\forall i \in N$, and $\forall f \in C$

$$g_4(\boldsymbol{x}) = \sum_{z=0}^{(b-minWD_{i,f}-1)} \max\left(\left(minWD_{i,f} - \sum_{j=z}^{(z+maxWD_{i,f})} \sum_{k=0}^{(r-1)} x_{i,(j \times r+k)}\right), 0\right). \tag{6}$$

S_5: Maximum number of consecutive free days is as follows: $\forall i \in N$, and $\forall f \in C$

$$g_5(\boldsymbol{x}) = \sum_{z=0}^{(b-maxFD_{i,f}-1)} \max\left(\left(\left(\sum_{j=z}^{(z+maxFD_{i,f})} \sum_{k=0}^{(r-1)} !x_{i,(j \times r+k)}\right) / r - maxFD_{i,f}\right), 0\right). \tag{7}$$

S_6: Minimum number of consecutive free days is as follows: $\forall i \in N$, and $\forall f \in C$

$$g_6(\boldsymbol{x}) = \sum_{z=0}^{(b-minFD_{i,f}-1)} \max\left(\left(minFD_{i,f} - \sum_{j=z}^{(z+minFD_{i,f})} \left(\sum_{k=0}^{(r-1)} !x_{i,(j \times r+k)} / r\right)\right), 0\right). \tag{8}$$

S_7: Assign complete weekends is as follows: $\forall i \in N$, and $\forall f \in C$

$$g_7(\boldsymbol{x}) = \sum_{w=0}^{(b/7-1)} \left(\sum_{j=(w \times 7+fstDay_{i,f})}^{(w \times 7+wkendDays_{i,f}+fstDay_{i,f}-1)} \sum_{k=0}^{r-1} !x_{i,(j \times r+k)}\right) \% wkendDays_{i,f}. \tag{9}$$

S_8: Assign identical complete weekends are as follows:

$$S_{8a} = \sum_{w=0}^{(b/7-1)} \sum_{i \in N} \sum_{f \in C} \sum_{k \in S} \left| x_{i,((w \times 7+fstDay_{i,f}) \times r+k)} - x_{i,((w \times 7+fstDay_{i,f}+1) \times r+k)} \right|$$

$$S_{8b} = \sum_{w=0}^{(b/7-1)} \sum_{i \in N} \sum_{f \in C} \sum_{k \in S} \left| x_{i,((w \times 7+fstDay_{i,f}) \times r+k)} - x_{i,((w \times 7+fstDay_{i,f}+1) \times r+k)+x_{i,((w \times 7+fstDay_{i,f}+2) \times r+k)}-1} \right|$$

---

[1] http://www.kuleuven-kortrijk.be/nrpcompetition.

Table 1 Notations used to formalize the INRC2010 datasets.

Indices	Description
b	Scheduling period (i.e., $b = 28$ days).
m	The total number of nurses.
r	The total number of shifts.
w	The total number of contracts.
q	The total number of skill categories.
u	The total number of unwanted patterns.
p	The total number of time slots $p = (b \times r)$.
$\boldsymbol{N}$	Set of nurses available in the dataset $\boldsymbol{N} = \{n_0, n_1, \ldots, n_{m-1}\}$.
$\boldsymbol{S}$	Set of shift types $\boldsymbol{S} = \{s_0, s_1, \ldots, s_{r-1}\}$.
$\boldsymbol{C}$	Set of contracts available for different nurses $\boldsymbol{C} = \{c_0, c_1, \ldots, c_{w-1}\}$.
$\boldsymbol{D}$	Set of days $\boldsymbol{D} = \{d_0, d_1, \ldots, d_{b-1}\}$.
$\boldsymbol{K}$	Set of skill categories $\boldsymbol{K} = \{k_0, k_1, \ldots, k_{q-1}\}$.
$\boldsymbol{T}$	Set of time sots $\boldsymbol{T} = \{t_0, t_1, \ldots, t_{p-1}\}$.
$\boldsymbol{PAT}$	Set of unwanted patterns $\boldsymbol{PAT} = \{pat_0, pat_1, \ldots, pat_{u-1}\}$.
$patLen_e$	The length of unwanted pattern pat_e.
$unPat_{e,s}$	Unwanted Pattern matrix: contains the details of each pattern pat_e at time period t_s.
$nurseSkill_{i,e}$	The skill category of nurse n_i is ke.
$shiftSkill_{k,e}$	The skill category ke is required for the shift s_k.
$dmnd_{j,k}$	Demand requirement of shift type s_k on day dj.
$maxSh_{i,f}$	Max number of shifts assigned for nurse n_i with contract c_f.
$minSh_{i,f}$	Min number of shifts assigned for nurse n_i with contract c_f.
$maxWD_{i,f}$	Max number of consecutive working days for nurse n_i with contract c_f.
$minWD_{i,f}$	Min number of consecutive working days for nurse n_i with contract c_f.
$maxFD_{i,f}$	Max number of consecutive free days for nurse n_i with contract c_f.
$minFD_{i,f}$	Min number of consecutive free days for nurse n_i with contract c_f.
$maxWW_{i,f}$	Max working weekend in four weeks for nurse n_i with contract c_f.
$wkendDays_{i,f}$	Number of days as weekend for nurse n_i with contract c_f.
$fstDay_{i,f}$	First day as weekend for nurse n_i with contract c_f.
$dayOff_{i,j}$	Day_Off matrix: whether the nurse n_i prefers not to work on day dj, $dayOff_{i,j} = \begin{cases} 1, & \text{if the nurse} n\_i \text{prefers not to work on day} d\_j, \\ 0, & otherwise. \end{cases}$
$dayOn_{i,j}$	Day_On matrix: whether the nurse n_i prefers to work on day dj, $dayOn_{i,j} = \begin{cases} 1, & \text{if the nurse} n\_i \text{prefers to work on day} d\_j, \\ 0, & otherwise. \end{cases}$
$shiftOff_{i,j}$	Shift_Off matrix: whether the nurse n_i prefers not to be assigned a specific shift s_k for day dj, $shiftOff\_i,j,k = \begin{cases} 1, & \text{if the nurse} n\_i \text{prefers to not assign specific shift } s\_k \text{for day} d\_j, \\ 0, & \text{otherwise.} \end{cases}$
$shiftOn_{i,j}$	Shift_On matrix: whether the nurse n_i prefers to be assigned a specific shift s_k for day dj, $shiftOn\_i,j,k = \begin{cases} 1, & \text{if the nurse} n\_i \text{prefers to assign specific shift } s\_k \text{for day} d\_j, \\ 0, & \text{otherwise.} \end{cases}$
$x_{i,j}$	is a binary variable, 1 if nurse n_i is assigned at time slot t_j, 0 otherwise.
$\boldsymbol{x}$	is a two-dimension solution roster $(m \times p)$.

$$g_8(\boldsymbol{x}) = \begin{cases} S_{8a} & wkendDays_{i,f} = 2, \\ S_{8b} & wkendDays_{i,f} = 3. \end{cases} \tag{10}$$

S_9: Two free days after a night shift is as follows: $\forall\ i \in N$, and y = index of night shift

$$g_9(\boldsymbol{x}) = \sum_{j=1}^{(b-2)} \max\left(\left(x_{i,((j-1)\times r+y))} - \sum_{k=0}^{(r-1)} x_{i,(j\times r+k)} + \sum_{k=0}^{(r-1)} x_{i,((j+1)\times r+k)} - 1\right), 0\right). \tag{11}$$

S_{10}: Requested day-Off is as follows: $\forall i \in N$

$$g_{10}(\boldsymbol{x}) = \sum_{j=0}^{(b-1)} \left(dayOff_{i,j} \quad \Lambda \quad \sum_{k=0}^{(r-1)} x_{i,(j\times r+k)} \right). \tag{12}$$

S_{11}: Requested day-On is as follows: $\forall i \in N$

$$g_{11}(\boldsymbol{x}) = \sum_{j=1}^{(b-1)} \left(dayOn_{i,j} \quad \Lambda \quad ! \sum_{k=0}^{(r-1)} x_{i,(j\times r+k)} \right). \tag{13}$$

S_{12}: Requested shift-Off is as follows: $\forall i \in N$

$$g_{12}(\boldsymbol{x}) = \sum_{j=0}^{(b-1)} \sum_{k\in S} \left(shiftOff_{i,j,k} \quad \Lambda \quad x_{i,(j\times r+k)} \right). \tag{14}$$

S_{13}: Requested shift-On is as follows: $\forall i \in N$

$$g_{13}(\boldsymbol{x}) = \sum_{j=0}^{(b-1)} \sum_{k\in S} \left(shiftOn_{i,j,k} \quad \Lambda \quad !x_{i,(j\times r+k)} \right). \tag{15}$$

S_{14}: Alternative skill is as follows: $\forall i \in N$, and $\forall e \in K$

$$g_{14}(\boldsymbol{x}) = \sum_{j=0}^{(b-1)} \sum_{k\in S} \left(x_{i,(j\times r+k)} \quad \Lambda \quad shiftSkill_{k,e} \quad \Lambda \quad nurseSkill_{i,e} \right). \tag{16}$$

S_{15}: Unwanted patterns are as follows:

$$S_{15}(\boldsymbol{x}) = \sum_{i\in N} \sum_{j\in D} \sum_{e\in PAT} \sum_{index=j}^{(patLen_e-1)} \sum_{k=0}^{K-1} \left(x_{i,(index\times r+k)} \ \Lambda \ unPat_{e,(index\times r+k)} \right).$$

$$g_{15}(\boldsymbol{x}) = \begin{cases} 1 & S_{15} = patLen_e, \\ 0 & S_{15} \neq patLen_e. \end{cases} \tag{17}$$

The nurse roster is evaluated using the objective function formalized in (18) that adds up the penalty of soft constraint violations in a feasible roster.

$$\min \quad f(\boldsymbol{x}) = \sum_{s=1}^{15} c_s . g_s(\boldsymbol{x}). \tag{18}$$

Note that s refers to the index of the soft constraint, 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 for the soft constraint s in solution roster $\boldsymbol{x}$.

3. Harmony Search Algorithm for NRP

The Harmony Search Algorithm (HSA) is an optimization method inspired by the musical improvisation process. Naturally, musicians play their instruments, practice by practice, seeking for a pleasing harmony (a perfect state) as determined by an audio-aesthetic standard. In optimization terms, the improvisation process is seeking for the (near-) optimal solution determined by an objective function. The pitch (= value) of each musical instrument (= decision variable) is part of aesthetic quality (= objection function) for the harmony.

HSA includes five main steps that will be described below. Algorithm 1 is the HSA pseudo-code for NRP.

Step1: *Initialize the parameters of the NRP and HSA*. The parameters of NRP are extracted from the raw data of the INRC2010 dataset, which includes for each nurse the maximum number of assignments; the minimum number of assignments; the maximum number of consecutive working days; the minimum number of consecutive working days; the maximum number of consecutive free days; the minimum of consecutive free days; the days of weekends; assigning complete weekend; assigning identical weekend; assigning two free days after night shift; defining the alternative skills if they exist; and defining the set of unwanted patterns. Furthermore, the nurse preferences parameters are drawn from the datasets that include day-Off, day-On, shift-Off and shift-On.

The roster is represented as a vector of allocations, i.e., $\boldsymbol{x} = (x_1, x_2 \ldots, x_E)$, where each allocation is a combination of four values (Nurse, Day, Shift, MCFlag) as shown in Table 2. MCFlag takes the value 1 when the allocation is assigned by the memory consideration operator or zero, otherwise. The length of roster $\boldsymbol{x}$ is $\boldsymbol{E}$ and is calculated as shown in (19). This roster should be evaluated by the objective function (18).

Table 2 Roster $\boldsymbol{x}$ representation.

allocation	Value			
	Nurse	Day	Shift	MCFlag
x_1	1	1	D	1
x_2	12	27	L	1
x_3	1	4	N	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
x_{E-1}	9	1	E	1
x_E	1	7	DH	1

Table 3 Ordering of shifts based on heuristic ordering method.

Shift type	Weekly nurses demand							Ordering
	Mon	Tue	Wed	Thu	Fri	Sat	Sun	
D	10	10	8	10	10	7	7	5
E	5	5	4	5	5	3	3	3
L	7	7	6	7	7	5	5	4
N	3	3	3	3	3	2	2	2
DH	1	1	1	1	1	1	1	1

$$E = \sum_{j=0}^{(b-1)} \sum_{k=0}^{(r-1)} dmnd_{j,k} \tag{19}$$

The control parameters of HSA are also initialized in this step, which includes the harmony memory size (**HMS**) to determine the number of rosters stored in the harmony memory (**HM**), the harmony memory consideration rate (**HMCR**) used in the improvisation process to determine the rate of selecting the allocations from HM rosters, the pitch adjusting rate (**PAR**) also used in the improvisation process to determine the probability of adjusting the allocations in a roster to neighboring allocations, and the maximum number of improvisations (**NI**) corresponding to the number of iterations.

Step2: *Initialize the harmony memory (HM)*. The HM is a space in memory used to keep the set of different rosters as determined by HMS (see (20)). The heuristic ordering (Burke et al., 2008) is used to construct the initial feasible rosters and store them in HM in ascending order based on the objective function value, where $f(\boldsymbol{x}^1) \leqslant f(\boldsymbol{x}^2) \leqslant \cdots \leqslant f(\boldsymbol{x}^{HMS})$.

$$\boldsymbol{HM} = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_N^1 & f(\boldsymbol{x}^1) \\ x_1^2 & x_2^2 & \cdots & x_N^2 & f(\boldsymbol{x}^2) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{x}_1^{HMS} & \boldsymbol{x}_2^{HMS} & \cdots & \boldsymbol{x}_N^{HMS} & f(\boldsymbol{x}^{HMS}) \end{bmatrix} \tag{20}$$

The procedure of assigning nurses to shifts by using heuristic ordering is carried out as follow: sorting the different shifts in ascending order based on the difficulty level, noting that the lowest weekly nurses demand is the highest difficulty (see Table 3). Then the required nurses of the ordered shifts will be assigned starting with the most difficult and ending with the easiest. Furthermore, the worst roster $\boldsymbol{x}^{worst}$ (i.e., the roster with the highest penalty value) in HM is defined.

Step3: *Improvise a new harmony roster*. In this step, the new roster $\boldsymbol{x}' = \left(x'_1, x'_2 \ldots, x'_E\right)$ is improvised based on three operators: (i) memory consideration, (ii) random consideration, and (iii) pitch adjustment. The feasibility of the new roster $\boldsymbol{x}'$ is considered during the improvisation process. If the improvisation process fails to improvise a feasible roster, the repair procedure will be triggered to maintain the feasibility of the new roster. The three operators work as follows:

- *Memory consideration*. This operator randomly selects a feasible value for the allocation x'_j in the new roster $\boldsymbol{x}'$ from the feasible set of alternative rosters stored in HM. In this paper, we improve HSA for NRP by replacing the random selection of this operator with a Global-best selection of Particle Swarm Optimization. The value of the allocation $\boldsymbol{x}'_j$ in the new roster $\boldsymbol{x}'$ will be assigned with the best value from the feasible set of alternative rosters stored in HM such as $x'_j \in \mathbf{R}_j$, where $\mathbf{R}_j = \left\{x^i_j | i = 1, 2, \ldots, HMS\right\}$ with probability (w.p.) of HMCR where HMCR $\in$ [0,1]. In other words, the allocation x'_j will be assigned the value of x^1_j if feasibility is achieved. If not, the value of the second alternative x^2_j will be assigned and so on until the last alternative x^{HMS}_j is reached. It is worth mentioning that when $\mathbf{R}_j = \phi$, this means that all alternatives have failed to come up with a feasible roster. In this case the *random consideration* operator will be triggered.

- *Random consideration*. This operator randomly selects a value for allocation x'_j from its feasible range X_j with a probability (1-HMCR) where the rules of heuristic ordering are considered. The *memory consideration* and *random consideration* operators select the value of x'_j as follows:

$$\boldsymbol{x}'_j \leftarrow \begin{cases} \mathbf{R}_j & \text{w.p.} \quad \text{HMCR}, \\ \mathbf{X}_j & \text{w.p.} \quad (1 - \text{HMCR}). \end{cases}$$

- *Pitch adjustment*. This operator adjusts the allocation x'_j selected by the memory consideration to its neighboring value during the improvisation process. In this paper, the pitch adjustment operator will be triggered when the improvisation process is completed rather than during the improvisation process. This is due to the fact that some of the soft constraints are not able to evade violation during the improvisation process. In other words, these constraints need a complete roster rather than a partial roster to evade violations such as $(S_1 - S_6)$.

This operator adjusts the allocation x'_j selected by the memory consideration (i.e., Memory Consideration Flag MCFlag $\left(x'_j\right)$ = true) to its neighboring value with probability PAR, where PAR $\in$ [0, 1], as follows:

$$\text{Pitch adjustment for} x'_j? \leftarrow \begin{cases} \text{Yes} & \text{w.p.} \quad \text{PAR}, \\ \text{NO} & \text{w.p.} \quad (1 - \text{PAR}). \end{cases}$$

For NRP, if the pitch adjustment decision for the allocation x'_j is 'Yes', one out of eight local changes will be triggered as follows:

$$x'_j \leftarrow \begin{cases} \mathit{MoveOneShift} & 0 \leqslant \mathit{rnd} \leqslant \text{PAR}/8, \\ \mathit{SwapOneShift} & \text{PAR}/8 < \text{rnd} \leqslant 2 \times \text{PAR}/8, \\ \mathit{TokenRingMove} & 2 \times \text{PAR}/8 < \mathit{rnd} \leqslant 3 \times \text{PAR}/8, \\ \mathit{Swap2Shifts} & 3 \times \text{PAR}/8 < \mathit{rnd} \leqslant 4 \times \mathit{PAR}/8, \\ \mathit{CrossMove} & 4 \times \text{PAR}/8 < \mathit{rnd} \leqslant 5 \times \text{PAR}/8, \\ \mathit{MoveWeekend} & 5 \times \text{PAR}/8 < \mathit{rnd} \leqslant 6 \times \text{PAR}/8, \\ \mathit{SwapConsecutive2Days} & 6 \times \text{PAR}/8 < \mathit{rnd} \leqslant 7 \times \text{PAR}/8, \\ \mathit{SwapConsecutive3Days} & 7 \times \text{PAR}/8 < \mathit{rnd} \leqslant \text{PAR}, \\ \mathit{DoNothing} & \text{PAR} < \mathit{rnd} \leqslant 1. \end{cases}$$

Where *rnd* is generated randomly between (0, 1). The eight pitch adjustment neighborhood structures are designed to run as follows:

Algorithm 1. The Harmony Search Algorithm for NRP

```
Step1  Initialize the parameters of NRP and HAS
    Set the NRP parameters drawn from the INRC2010 dataset.
    Set the HSA parameters (HMCR, PAR, NI, HMS).
    Define the roster representation and utilize the objective function.
Step2  Initialize the harmony memory
    Construct rosters of the harmony memory by using heuristic ordering method and store them in an ascending order, HM = $\{x^1, x^2, \ldots, x^{HMS}\}$.
    Identify the worst roster in HM, $x^{worst} = x^{HMS}$.
Step3  Improvise a new harmony
    $x' = \phi$        {$x'$ is the new roster
    for each $j \in [1,E]$ do
        if (U(0,1) $\leqslant$HMCR) then
            $x'_j \in R_j$, where
            $R_j = \left\{x^i_j | i = 1, 2, \ldots, HMS\right\}$        {memory consideration operator}
            if $R_j = \phi$ then
                $x'_j \in X_j$        {random consideration operator}
            else
                MCFlag$\left(x'_j\right)$ = true
            end if
        else
            $x'_j \in X_j$        {random consideration operator}
        end if
    end for
    for each $j \in [1,E]$ do
        if MCFlag$\left(x'_j\right)$ = true then
            if (U(0,1) $\leqslant$PAR) then
                pitch adjustment for $\left(x'_j\right)$        {pitch adjustment operator}
            end if
        end if
    end for
Step4  Update the harmony memory
    if ($f(x') < f(x^{worst})$) then
        Replaces $x^{worst}$ by $x'$ in the HM.
        Reordering the rosters in HM in an ascending order.
    end if
Step5  Check the stop criterion
    while (the maximum number of improvisations NI is not reached) do
        Repeat Step3 to Step5
    end while
```

1. *MoveOneShift pitch adjustment*. The nurse of the selected allocation x'_j will be replaced by another nurse selected randomly to decrease the penalty of different soft constraint violations with probability [0, PAR/8].
2. *SwapOneShift pitch adjustment*. The shift of selected allocation x'_j will be exchanged with another shift with another nurse on the same day for another selected allocation x'_k with probability (PAR/8, PAR/4].
3. *TokenRingMove pitch adjustment*. The nurse of selected allocation x'_j will be replaced by another nurse selected randomly if the soft constraint S_7 is violated. Furthermore, the shift of a selected allocation x'_j will be exchanged with

another shift on which another nurse is working on the same day, for another selected allocation x'_k to solve the violation of the soft constraint S_8. This pitch adjustment procedure is triggered with probability (PAR/4, 3 × PAR/8].

4. *Swap2Shifts pitch adjustment*. The shift of selected allocation x'_j will be exchanged with another shift having another nurse on the same day for another selected allocation x'_k, and selects the third allocation x'_q with the same nurse and different day of x'_j, and the same shift of x'_k to be exchanged with another shift for the same nurse of x'_k, day of x'_q, and shift of x'_j for the fourth selected allocation x'_r with probability (3 × PAR/8, PAR/2].
5. *CrossMove pitch adjustment*. The day of a selected allocation x'_j will be exchanged with another day with another nurse and the same shifts for another selected allocation x'_k with probability (PAR/2, 5 × PAR/8].
6. *MoveWeekend pitch adjustment*. If the day of a selected allocation x'_j is a weekend day, then the nurse of x'_j and all weekend allocations will be moved to another nurse selected randomly with probability (5 × PAR/8, 6 × PAR/8].
7. *SwapConsecutive2Days pitch adjustment*. This pitch adjustment is made to move a group of shifts of two consecutive days among nurses. The nurse of a selected allocation x'_j and the other allocation x'_k, where the two allocations for the same nurse and the day of x'_k is the next or previous day of x'_j will be exchanged with another nurse selected randomly with probability (6 × PAR/8, 7 × PAR/8].
8. *SwapConsecutive3Days pitch adjustment*. This pitch adjustment is designed to move a group of shifts of three consecutive days among nurses. The nurse of a selected allocation x'_j and the other two allocations x'_k and x'_q, where the three allocations for the same nurse and the days of x'_k and x'_q are the next or previous day of x'_j will be exchanged with another nurse selected randomly with probability (7 × PAR/8, PAR].

In this paper, any local changes that do not improve the new roster, or result in an unfeasible roster, will be discarded. It is worth noting that when the improvisation process is completed by using the memory consideration and random consideration operators, the new roster is tested for completion (i.e., all allocations are assigned with values). If not complete, the repair process will be triggered to fulfill unassigned allocations with feasible values. The repair process consists of three steps: first, identify all allocations that are not scheduled in the new roster; second, identify the day(s) where the nurses demand are not completely scheduled in the new roster; and third, for each day identified, copy the allocations of the same day from the previous or next week.

Step4: Update the harmony memory. After a new roster $\boldsymbol{x}'$ is improvised, the HM will be updated by the "survival of the fittest" between the new roster and the worst roster $\boldsymbol{x}^{worst}$ in HM. That is, the new roster $\boldsymbol{x}'$ replaces the worst roster $\boldsymbol{x}^{worst}$ in HM. Furthermore, reordering the rosters in HM in an ascending order will be considered.

Step5: Check the stop criterion. Based on NI (maximum number of improvisation), *Step3* to *Step5* of HSA are repeated.

4. Illustrative example of applying HSA for NRP

Table 4 shows an illustrative example of a nurse roster. The roster includes the different schedules of four nurses for one week scheduling period. Each *row* represents a schedule of a nurse in the roster, each *column* represents a day, and each filled cell contains the shift type assigned to a nurse. It is worth mentioning that two types of shifts *D* for day shift and *L* for Late shift are available.

Table 4 Illustrative example of feasible Nurse Roster.

n_0	D	D	D	L	L		
n_1		L	L			L	L
n_2	L		D	D	D		
n_3	D	D		D	D	D	D

4.1. Initialize the parameters of the NRP and HSA

The nurse roster in Table 4 includes assigning two shifts $\boldsymbol{S} = \{D, L\}$ for four nurses $\boldsymbol{N} = \{n_0,n_1,n_2,n_3\}$ over seven days scheduling period $\boldsymbol{D} = \{d_0,d_1, \ldots , d_6\}$. This is a feasible roster which is mapped to the vector $\boldsymbol{x} = (x_1,x_2 \ldots ,x_{19})$, where 19 is the number of assignments in the roster. The allocation x_1 takes a map value of (Nurse, Day, Shift, MCFlag). Furthermore, the parameters of HSA are initialized as NI = 1000, HMS = 5, HMCR = 0.99, and PAR = 0.1. The rosters $\boldsymbol{x}$ are evaluated using the objective function (see (18)).

4.2. Initialize the harmony memory

Table 5 shows the rosters in harmony memory that are generated using a heuristic ordering method as many as HMS. Note that the rosters in harmony memory are sorted in an ascending order in accordance with their objective function values (see the last column of Table 5).

4.3. Improvise a new harmony roster

In this step, the new nurse roster $\boldsymbol{x}'$ is improvised based on three operators' memory consideration (MC), random consideration (RC), and pitch adjustment (PA) as shown in Table 6. Then, the improvisation process is performed and evaluated with the objective function. Assuming the value of the objective function $f(\boldsymbol{x}') = 170$.

4.4. Update the harmony memory

Apparently, the objective function value of the new roster $\boldsymbol{x}'$ is better than that of x^{worst} in HM (i.e., $f(\boldsymbol{x}') < f(x^5)$). Thus, the new roster replaces the worst one in HM and is re-sorted according to the objective function value as shown in Table 7.

4.5. Check the stop criterion

The iterative process of Steps 4.3–4.4 in HSA is performed for NI = 1000 iterations.

5. Experimental results

The proposed Harmony Search Algorithm is programmed using Microsoft Visual C++ 6.0, under windows Vista, on an Intel Machine with CoreTM processor 2.66 GHz, and 4 GB RAM. The dataset introduced by the INRC2010 for

Table 5 Harmony Memory restores.

	x_1	x_2	x_3	…	x_{18}	x_{19}	$f(\mathbf{x})$
$\mathbf{x}^1$	$(n_0,d_0,D,0)$	$(n_2,d_0,L,0)$	$(n_3,d_0,D,0)$	…	$(n_1,d_6,L,0)$	$(n_3,\ d_6,D,0)$	287
$\mathbf{x}^2$	$(n_0,d_0,D,0)$	$(n_1,d_0,D,0)$	$(n_2,d_0,L,0)$	…	$(n_0,d_6,D,0)$	$(n_3,d_6,L,0)$	301
$\mathbf{x}^3$	$(n_2,d_0,D,0)$	$(n_0,d_0,L,0)$	$(n_3,d_0,D,0)$	…	$(n_1,d_6,L,0)$	$(n_2,d_6,D,0)$	311
$\mathbf{x}^4$	$(n_1,d_0,D,0)$	$(n_3,d_0,D,0)$	$(n_0,d_0,L,0)$	…	$(n_1,d_6,D,0)$	$(n_0,d_6,L,0)$	325
$\mathbf{x}^5$	$(n_3,d_0,D,0)$	$(n_1,d_0,L,0)$	$(n_0,d_0,D,0)$	…	$(n_2,d_6,L,0)$	$(n_3,d_6,D,0)$	450

Table 6 Improvising the new roster $\mathbf{x}'$.

	MC	RC	PA	Results
x'_1	√	–	–	$(n_0,d_0,D,1)$
x'_2	–	√	–	$(n_1,d_0,D,0)$
x'_3	√	–	√	$(n_2,d_0,\ L,1) \rightarrow (n_3,d_0,L,1)$
⋮	⋮	⋮	⋮	⋮
x'_{18}	√	–	–	$(n_1,d_6,L,1)$
x'_{19}	√	–	√	$(n_3,d_6,\ D,1) \rightarrow (n_1,d_6,D,1)$

Nurse Rostering is used for studying the effectiveness of HSA proposed for NRP.

5.1. INRC2010 dataset

The dataset established by INRC2010 is classified into three tracks: *sprint*, *medium*, and *long* datasets based on complexity and size. Each track is categorized into four types in accordance with their publication time with reference to the competition: early, late, hidden, and hint.

The *sprint track* includes 33 datasets, which consist of 10 early, 10 late, 10 hidden, and 3 hint. These datasets are the easiest, including 10 nurses with one skill qualification and 3–4 different contract types, and the daily shifts are 4 for 28 days scheduling period. The *medium track* includes 18 datasets, which are categorized as 5 early, 5 late, 5 hidden, and 3 hint. These datasets are more complicated than the sprint track datasets, including 30–31 nurses with 1 or 2 skills and 4 or 5 different contracts. The daily shifts are 4 or 5 shifts over 28 days scheduling period. The *long track* includes 18 datasets, which are categorized as 5 early, 5 late, 5 hidden, and 3 hint. These datasets are the hardest, including 49–50 nurses with 2 skills and 3 or 4 different contracts. The daily shifts are 5 shifts for 28 days scheduling period.

Tables 8–10 include the different characteristics of sprint, medium, and long track datasets, respectively, where the combination of ''Type'' and ''Index'' columns is used to label the dataset. The ''Shift'' represents the number of shifts, ''Contracts'' is for the number of the contracts available in the dataset, and ''Unwanted'' is for the number of unwanted patterns. The number of days during the weekends is indicated by the ''Weekend'' column. The existence of nurse preferences: day-

Table 7 Updated harmony memory.

	x_1	x_2	x_3	…	x_{18}	x_{19}	$f(\mathbf{x})$
$\mathbf{x}^1$	$(n_0,d_0,D,1)$	$(n_1,d_0,D,0)$	$(n_3,d_0,L,1)$	…	$(n_1,d_6,L,1)$	$(n_1,\ d_6,D,1)$	170
$\mathbf{x}^2$	$(n_0,d_0,D,0)$	$(n_2,d_0,L,0)$	$(n_3,d_0,D,0)$	…	$(n_1,d_6,L,0)$	$(n_3,\ d_6,D,0)$	287
$\mathbf{x}^3$	$(n_0,d_0,D,0)$	$(n_1,d_0,D,0)$	$(n_2,d_0,L,0)$	…	$(n_0,d_6,D,0)$	$(n_3,\ d_6,L,0)$	301
$\mathbf{x}^4$	$(n_2,d_0,D,0)$	$(n_0,d_0,L,0)$	$(n_3,d_0,D,0)$	…	$(n_1,d_6,L,0)$	$(n_2,\ d_6,D,0)$	311
$\mathbf{x}^5$	$(n_1,d_0,D,0)$	$(n_3,d_0,D,0)$	$(n_0,d_0,L,0)$	…	$(n_1,d_6,D,0)$	$(n_0,\ d_6,L,0)$	325

Table 8 Sprint track dataset characteristics.

Type	Index	Shifts	Skills	Contracts	Unwanted	Weekend	Day Off	Shift Off	Period
Early	01–10	4	1	4	3	2	√	√	1–28/01/2010
Hidden	01–02	3	1	3	4	2	√	√	1–28/06/2010
	03, 05, 08	4	1	3	8	2	√	√	1–28/06/2010
	04, 09	3, 4	1	3	8	2	√	√	1–28/06/2010
	06–07	3	1	3	4	2	√	√	1–28/01/2010
	10	4	1	3	8	2	√	√	1–28/01/2010
Late	01, 03–05	4	1	3	8	2	√	√	1–28/01/2010
	02	3	1	3	4	2	√	√	1–28/01/2010
	06–07, 10	4	1	3	0	2	√	√	1–28/01/2010
	08	4	1	3	0	2	*X*	*X*	1–28/01/2010
	09	4	1	3	0	2, 3	*X*	*X*	1–28/01/2010
Hint	01, 03	4	1	3	8	2	√	√	1–28/01/2010
	02	4	1	3	0	2	√	√	1–28/01/2010

Table 9 Medium track dataset characteristics.

Type	Index	Shifts	Skills	Contracts	Unwanted	Weekend	Day Off	Shift Off	Period
Early	01–05	4	1	4	0	2	√	√	1–28/01/2010
Hidden	01–04	5	2	4	9	2	***X***	*X*	1–28/06/2010
	05	5	1	4	9	2	***X***	*X*	1–28/06/2010
Late	01	4	1	4	7	2	√	√	1–28/01/2010
	02, 04	4	1	3	7	2	√	√	1–28/01/2010
	03	4	1	4	0	2	√	√	1–28/01/2010
	05	5	2	4	7	2	√	√	1–28/01/2010
Hint	01, 03	4	1	4	7	2	√	√	1–28/01/2010
	02	4	1	3	7	2	√	√	1–28/01/2010

Table 10 Long track dataset characteristics.

Type	Index	Shifts	Skills	Contracts	Unwanted	Weekend	Day Off	Shift Off	Period
Early	01–05	5	2	3	3	2	√	√	1–28/01/2010
Hidden	01–04	5	2	3	9	2, 3	***X***	*X*	1–28/06/2010
	05	5	2	4	9	2, 3	***X***	*X*	1–28/06/2010
Late	01, 03, 05	5	2	3	9	2, 3	***X***	*X*	1–28/01/2010
	02, 04	5	2	4	9	2, 3	***X***	*X*	1–28/01/2010
Hint	01	5	2	3	9	2, 3	***X***	*X*	1–28/01/2010
	02, 03	5	2	3	7	2	***X***	*X*	1–28/01/2010

Table 11 Different cases to study effectiveness of proposed HSA.

Cases	HMS	HMCR	PAR
Case$_1$	10	0.99	0.1
Case$_2$	30	0.99	0.1
Case$_3$	50	0.99	0.1
Case$_4$	10	0.90	0.1
Case$_5$	10	0.95	0.1
Case$_1$	10	0.99	0.1
Case$_6$	10	0.99	0.0
Case$_1$	10	0.99	0.1
Case$_7$	10	0.99	0.4
Case$_8$	10	0.99	0.7

Off and shift-Off refer to “Day Off” and “Shift Off”, and the last column is for the scheduling period.

5.2. Experimental design

A series of experiments is carried out to evaluate the proposed HSA. In this paper, eight experimental cases are used to study the effectiveness of the proposed method. Each case has different values of parameter settings as shown in Table 11. Each experimental case is replicated 10 times for each dataset with the most suitable iteration numbers fixed to 100,000 for all runs. The first three cases are being used to study the effectiveness of the HSA with different HMS values (i.e., 10, 30, and 50). Case$_4$, Case$_5$, and Case$_1$ are employed to study the effectiveness of the proposed method with different HMCR values (i.e., 0.90, 0.95, and 0.99). The last four cases are used to find the best value of PAR for local improvement, and this is done for the purpose of studying the effect of local changes proposed in this paper on the HSA behaviour. In order to study the effect of Global-best memory consideration on the behavior of HSA, the case that obtained the best results will be run with the random selection to identify the power of Global-best memory consideration on the HSA behavior.

5.3. Experimental results and discussions

The results of the eight experimental cases, defined previously, are summarized in Tables 12–24 for the three tracks: *sprint*,*medium* and *long* datasets, respectively. Note that the numbers in the tables refer to the penalty values of soft constraint violations (lowest is best). For each dataset on each experimental case, the best (B.), mean (M.), worst (W.), and standard deviation (Std.) of 10 runs are recorded. The best result among all experimental cases on each dataset is highlighted in bold.

5.3.1. Studying the effects of HMS

The *HMS* parameter is studied in Case$_1$ to Case$_3$ with different HMS values (i.e., 10, 30, and 50), and the results are summarized in Tables 12–14. The HMS parameter represents the problem search space covered during the search, where the problem search space includes all possible solutions for the problem. The HMS with small value indicates a small number of solutions covered during the search with high speed of convergence. In contrast, the big value of HMS indicates a high number of solutions stored in HM, but with slow speed of convergence. Experimentally, the HMS with small values achieved the best results in most of the datasets, especially for medium and long datasets. Notably, the HMS = 10 shall be used in next cases.

5.3.2. Studying the effects of HMCR

The performance of HSA using different *HMCR* values is investigated in Case$_4$, Case$_5$, and Case$_1$. Tables 15–17 show

Table 12 The performance of HMS parameter settings for *sprint track* dataset.

Dataset	HMS = 10				HMS = 30				HMS = 50			
	B.	M.	W.	Std.	B.	M.	W.	Std.	B.	M.	W.	Std.
Early01	**61**	64.1	69	2.4	**61**	64.2	67	2.3	62	65.8	70	2.9
Early02	**64**	67.6	76	3.6	65	68.9	73	2.4	66	68.3	71	1.8
Early03	57	61.2	65	2.5	**56**	62.7	65	3.0	59	62.3	66	2.4
Early04	**68**	71.1	77	3.1	69	72.5	79	2.5	69	71.7	75	1.9
Early05	63	65.5	69	1.8	**62**	63.9	68	1.9	64	65.9	68	1.3
Early06	**58**	62.6	69	3.7	**58**	62.5	65	2.2	**58**	61.7	67	3.2
Early07	63	65.2	67	1.5	**62**	65.9	68	2.2	64	66.7	70	2.0
Early08	**59**	63.7	68	2.8	63	65.7	70	2.1	63	65.2	68	1.8
Early09	61	66.2	69	2.6	61	65.4	69	2.5	**60**	65.6	70	3.2
Early10	58	60.5	63	1.6	**55**	61.7	68	3.6	59	63	67	2.4
Late01	53	57.6	64	3.8	55	57.3	61	2.0	**50**	57	64	5.1
Late02	57	61.5	69	3.9	57	60.7	66	3.6	**54**	60.6	65	4.0
Late03	63	69.9	75	4.3	**55**	64.5	71	4.6	66	69.8	78	3.9
Late04	**112**	125.4	152	12	117	128.1	143	9.2	**112**	129.6	146	9.9
Late05	**55**	62.3	72	5	57	62.3	66	2.9	57	63.8	73	6.3
Late06	**51**	55.6	60	3.3	53	56.3	59	2.2	**51**	56.8	63	4.1
Late07	**60**	74.1	86	8.4	65	74.4	90	8.4	68	76.8	91	7.0
Late08	**21**	32.8	40	6.4	23	36.1	62	11.7	27	40.8	53	7.2
Late09	**28**	40.9	60	10.3	30	37.8	45	6.2	29	36.8	48	5.8
Late10	61	80.2	96	11.1	65	80	96	8.5	**60**	74	87	9.0
Hidden01	**48**	54	65	4.9	52	55.5	63	4.0	54	58.5	66	3.7
Hidden02	**47**	53.9	58	4.3	**47**	56.6	66	5.2	52	57.6	64	3.9
Hidden03	**78**	88.1	99	6.5	80	86.5	95	5.3	79	89.2	96	5.0
Hidden04	80	86.2	90	3.7	**79**	87.7	94	4.3	80	88.9	99	5.4
Hidden05	**73**	80.8	88	5	78	84.1	91	4.4	76	85.6	91	5.0
Hidden06	207	230	280	20.4	215	237.4	269	16.4	**202**	228.3	249	16.4
Hidden07	**196**	262.5	307	33.8	218	254.8	301	22.6	230	266	309	24.6
Hidden08	**267**	294.3	327	19.8	278	303.8	320	15.1	274	318.1	359	25.5
Hidden09	**373**	412.7	442	24.4	383	424.9	444	20.6	400	435.5	468	17.5
Hidden10	**346**	412.3	467	40.1	385	434.9	474	29.7	399	435.4	530	43.5
Hint01	**101**	120.4	136	10.7	111	126	152	12.7	**101**	121.5	148	11.4
Hint02	**59**	75.6	94	11.1	68	75.9	83	4.4	64	77.3	89	8.5
Hint03	84	97.6	108	8.2	91	107.5	130	12.2	**78**	111.6	132	14.3

Table 13 Performance of HMS parameter settings for *medium track* dataset.

Dataset	HMS = 10				HMS = 30				HMS = 50			
	B.	M.	W.	Std.	B.	M.	W.	Std.	B.	M.	W.	Std.
Early01	**328**	338.5	354	8.2	345	360.4	373	9.2	353	370.1	383	7.9
Early02	**321**	334.8	352	9	347	360.9	371	7.0	352	364.7	379	10.0
Early03	**318**	329.9	344	8.5	331	349.8	365	9.4	337	353.7	370	10.0
Early04	**314**	332.1	342	8.9	342	357.8	366	8.5	350	372.2	389	10.3
Early05	**386**	399.9	412	8.8	413	424.5	431	5.1	433	441.7	448	5.7
Late01	**366**	411.8	461	35.9	495	542.3	634	43.4	562	634.4	748	59.5
Late02	**104**	126.3	145	12.6	152	167.3	184	10.6	189	198.8	217	9.2
Late03	**116**	141.6	153	11.2	167	186.7	202	13.1	215	232.3	252	13.1
Late04	**103**	121.6	132	10	158	174.4	185	9.4	181	195.5	208	9.6
Late05	**391**	421.9	471	29	525	593.7	653	39.3	617	705.3	795	68.8
Hidden01	**460**	514.1	615	47.2	742	818.9	899	58.8	802	932.1	1050	84.2
Hidden02	**551**	604.2	659	38.8	683	787.7	895	74.0	736	898.4	976	76.7
Hidden03	**158**	171.7	188	11.6	215	243.7	265	16.0	238	270.1	298	18.8
Hidden04	**208**	224.7	257	13.9	274	292.8	310	12.2	298	324.5	351	17.0
Hidden05	**455**	502.8	555	37.1	576	718.3	822	70.7	743	861.3	966	88.7
Hint01	**134**	158.5	183	15.8	202	226.1	240	10.9	233	257.3	270	11.5
Hint02	**297**	357	404	34.2	513	573.4	705	65.6	534	643.3	713	47.9
Hint03	**315**	445	536	63.2	729	867.7	1053	119.0	891	998.5	1163	105.6

Table 14 Performance of HMS parameter settings for *long track* dataset.

Dataset	HMS = 10				HMS = 30				HMS = 50			
	B.	M.	W.	Std.	B.	M.	W.	Std.	B.	M.	W.	Std.
Early01	**350**	362.2	377	9.7	367	386.5	430	20.0	369	393.6	414	12.8
Early02	**381**	412.3	435	16.5	417	425.6	436	5.5	417	445.8	466	14.8
Early03	**356**	372.2	394	12.5	369	389.4	403	10.0	379	401	421	14.5
Early04	**454**	466.3	484	8.9	467	487.1	504	10.9	491	502.8	511	7.1
Early05	**422**	439.2	458	9.7	443	467.1	479	12.0	459	473.7	483	7.0
Late01	**1355**	1481	1607	89.7	1401	1723.7	2016	191.7	1745	1822.4	1941	65.1
Late02	**1297**	1518.1	1704	121.1	1650	1825.8	2065	138.9	1657	1916.1	2253	175.3
Late03	**1288**	1521.3	1639	115.4	1602	1703.2	1837	73.7	1636	1837.3	2007	121.3
Late04	**1385**	1503.3	1664	91.1	1585	1736.8	1927	121.9	1713	1926.2	2102	112.0
Late05	**993**	1164	1337	106.4	1248	1402.1	1538	85.1	1387	1476.9	1552	49.6
Hidden01	**1590**	1651.8	1728	51.1	1851	1966.2	2164	117.6	1887	2057.6	2265	124.6
Hidden02	**401**	443.6	484	28.3	482	505.4	537	16.9	494	520.6	550	18.2
Hidden03	**274**	330.4	367	32.7	370	392.3	424	18.8	402	434.8	475	22.5
Hidden04	**310**	336.7	368	19	364	391.9	451	24.9	401	426.9	463	19.2
Hidden05	**296**	362	431	35.1	392	418.7	466	23.3	412	453	485	24.2
Hint01	**338**	368.8	421	25.2	389	424.1	457	24.4	420	454	483	18.2
Hint02	**235**	262.7	284	13	280	307.3	328	15.2	302	317.4	336	12.9
Hint03	**1040**	1173.9	1413	107.7	1179	1291.4	1449	81.8	1252	1446.6	1712	125.1

Table 15 The performance of HMCR parameter settings for *sprint track* dataset.

Dataset	HMCR = 0.90				HMCR = 0.95				HMCR = 0.99			
	B.	M.	W.	Std.	B.	M.	W.	Std.	B.	M.	W.	Std.
Early01	103	111.1	119	5.5	81	86.9	97	5.1	**61**	64.1	69	2.4
Early02	103	111.8	127	7.3	81	87.4	103	6.6	**64**	67.6	76	3.6
Early03	102	110.2	123	6.7	77	83.2	88	4.4	**57**	61.2	65	2.5
Early04	110	120.4	140	8.7	91	96	105	4.1	**68**	71.1	77	3.1
Early05	106	113.4	121	4.6	77	85.3	98	6.6	**63**	65.5	69	1.8
Early06	102	105.9	111	3.1	76	81.2	89	4.7	**58**	62.6	69	3.7
Early07	102	112.5	121	6.3	78	89.2	96	5.4	**63**	65.2	67	1.5
Early08	102	108.1	113	3.7	78	83.8	92	4.2	**59**	63.7	68	2.8
Early09	113	118.9	122	3.1	80	88	96	5.4	**61**	66.2	69	2.6
Early10	108	114.2	126	6.0	73	84	87	4.0	**58**	60.5	63	1.6
Late01	122	137.7	150	7.6	82	92.6	103	7.8	**53**	57.6	64	3.8
Late02	122	127.7	132	3.6	82	93.8	101	5.7	**57**	61.5	69	3.9
Late03	139	150.3	161	7.6	100	108.7	123	6.9	**63**	69.9	75	4.3
Late04	476	494.7	522	15.6	268	288.3	321	19.3	**112**	125.4	152	12
Late05	128	140	158	9.0	88	98.8	109	6.8	**55**	62.3	72	5
Late06	106	117.1	126	6.2	75	81.9	89	4.4	**51**	55.6	60	3.3
Late07	210	267.8	347	34.3	123	149	166	13.3	**60**	74.1	86	8.4
Late08	212	271.1	373	49.1	93	126	189	26.8	**21**	32.8	40	6.4
Late09	202	285.9	356	48.5	86	125.4	177	24.3	**28**	40.9	60	10.3
Late10	251	294.4	345	32.1	130	156.7	183	17.8	**61**	80.2	96	11.1
Hidden01	145	161	181	12.0	100	105.6	114	4.6	**48**	54	65	4.9
Hidden02	122	132.3	147	7.1	71	85.3	93	6.8	**47**	53.9	58	4.3
Hidden03	177	188.7	199	7.8	122	138.2	146	7.1	**78**	88.1	99	6.5
Hidden04	175	185.8	198	8.0	121	139.7	147	7.7	**80**	86.2	90	3.7
Hidden05	170	**186**	197	7.7	122	130.7	147	7.7	**73**	80.8	88	5
Hidden06	752	899.4	969	69.8	432	498.1	590	51.6	**207**	230	280	20.4
Hidden07	669	751.7	875	65.8	414	483.3	522	41.8	**196**	262.5	307	33.8
Hidden08	843	888.8	944	33.0	497	605.3	714	67.0	**267**	294.3	327	19.8
Hidden09	939	1031.4	1099	43.6	623	718.7	775	52.5	**373**	412.7	442	24.4
Hidden10	935	1020.9	1101	56.8	666	715.4	798	43.8	**346**	412.3	467	40.1
Hint01	372	420.3	446	23.5	226	253.7	286	18.0	**101**	120.4	136	10.7
Hint02	241	287.7	352	31.0	119	151.9	184	22.2	**59**	75.6	94	11.1
Hint03	415	457.8	509	33.5	240	262.5	289	13.8	**84**	97.6	108	8.2

Table 16 Performance of HMCR parameter settings for *medium track* dataset.

Dataset	HMCR = 0.90				HMCR = 0.95				HMCR = 0.99			
	B.	M.	W.	Std.	B.	M.	W.	Std.	B.	M.	W.	Std.
Early01	536	545.5	556	6.5	480	497.1	520	10.8	**328**	338.5	354	8.2
Early02	530	540.2	554	7.2	485	497	506	6.9	**321**	334.8	352	9
Early03	509	524.6	538	9.6	471	483.2	499	8.4	**318**	329.9	344	8.5
Early04	530	541.8	550	7.5	488	499.5	513	8.1	**314**	332.1	342	8.9
Early05	599	606	615	5.4	548	567.3	580	8.7	**386**	399.9	412	8.8
Late01	2066	2134.5	2186	37.6	1610	1712	1814	66.9	**366**	411.8	461	35.9
Late02	506	526.3	540	11.6	427	452.5	475	14.8	**104**	126.3	145	12.6
Late03	670	685.8	708	11.4	532	559.3	607	24.8	**116**	141.6	153	11.2
Late04	536	545.3	560	7.6	439	462.2	491	17.0	**103**	121.6	132	10
Late05	2051	2214.9	2394	93.7	1599	1757.9	1840	71.2	**391**	421.9	471	29
Hidden01	2997	3163.9	3258	76.3	2467	2583.6	2754	101.6	**460**	514.1	615	47.2
Hidden02	2386	2502.7	2601	70.9	1924	2091.2	2187	76.9	**551**	604.2	659	38.8
Hidden03	749	781.3	807	17.2	621	649.3	682	19.7	**158**	171.7	188	11.6
Hidden04	720	745.7	778	21.1	632	651.6	667	10.6	**208**	224.7	257	13.9
Hidden05	2648	2883.5	3022	106.0	2201	2373.4	2557	99.5	**455**	502.8	555	37.1
Hint01	730	754	793	17.5	582	628.4	681	27.1	**134**	158.5	183	15.8
Hint02	2361	2448.9	2511	50.0	1821	1973.1	2163	93.7	**297**	357	404	34.2
Hint03	5251	5941.5	6602	397.1	3458	3954.3	4331	277.2	**315**	445	536	63.2

the results of HSA with differing HMCR values (i.e., 0.90, 0.95, and 0.99). The HMCR parameter with high value leads to a higher exploitation and a lower exploration, and vice-versa. In other words, the higher value of HMCR parameter indicates intensively using the HM in the improvisation process during the search. The HMCR = 0.99 is recommended to solve the NRP based on achieving the best results in comparison with the other HMCR values.

5.3.3. Studying the effects of PAR

The performance of the HSA using different PAR values (i.e., 0.0, 0.1, 0.4, and 0.7) is studied in Case$_6$, Case$_1$, Case$_7$, and Case$_8$. The results of the four cases are summarized in Tables 18–20. The value of PAR represents the percentage of enhancing the solution locally by the different pitch adjustment procedures. The PAR with zero value indicates that the local search procedures are not used during the search. In other words, the solution is not locally enhanced. Experimentally, the PAR with high value (i.e., Case$_8$) obtained the best results in comparison with the other cases of PAR. This is due to the considerable local changes made in each iteration. Furthermore, the results of case! without the pitch adjustment operator perform poorly in comparison with other cases, either little usage of the pitch adjustment operator in Case$_6$, or intensive usage like in Case$_8$.

5.3.4. Studying the effects of Global-best

The effectiveness of the Global-best idea of the HSA is studied by running the case that achieved the best results (i.e., Case$_8$) using the original random selection of HSA. The results of the Global-best HSA and original HSA are summarized in Tables 21–23. Notably, the Global-best HSA achieved better results than the original HSA in most of the datasets, especially in *medium* and *long* track datasets. However, random selection is able to overcome the Global-best selection in *sprint* track dataset results. In contrast, the convergence speed of Global-best is faster than the original HSA, by virtue of the Global-best power to inherit the values of the allocations from the best rosters in HM in the process of improvising the new roster.

Fig. 1 shows the best results in HM of Global-best and random selection methods in each iteration for *long_hidden03* dataset. Note that, in this figure, 10000 iterations are used to show the distribution among the results visually. The color lines in this figure show the correlation between the number of iterations and the objective function value. These lines represent the best results in HM in each iteration. An analysis of the diagram shows that the objective function value decreases as the number of iterations increases. Apparently, the slope of the Global-best selection is more than the random selection, especially at the beginning of the search.

5.4. Comparison with INRC2010 winners

This section compares the results produced by the proposed HSA with those produced by the winners' methods in INRC2010. The key for the winners' methods is shown in Table 24.

Tables 25–27 show the best results produced by the proposed method for 69 datasets published on the INRC2010 website, and compared with the five winners' methods of INRC2010. These results are the best results summarized in Tables 12–23 for all cases. Furthermore, these tables include the best results obtained by the winners' methods in INRC2010. Note that Table 19 includes the results of the Modified Harmony Search Algorithm (MHSA) presented in (Awadallah et al., 2011a). Basically, the proposed HSA was able to produce the best results for two datasets as achieved by the other winners' methods. In addition, the proposed HSA produced com-

Table 17 Performance of HMCR parameter settings for *long track* dataset.

Dataset	HMCR = 0.90				HMCR = 0.95				HMCR = 0.99			
	B.	M.	W.	Std.	B.	M.	W.	Std.	B.	M.	W.	Std.
Early01	624	648.3	665	12.0	565	578.5	596	10.3	**350**	362.2	377	9.7
Early02	676	693.1	707	8.3	620	630.9	646	7.8	**381**	412.3	435	16.5
Early03	626	634.5	650	7.8	562	576.7	584	7.1	**356**	372.2	394	12.5
Early04	730	744.9	762	9.3	660	685.6	711	15.3	**454**	466.3	484	8.9
Early05	713	728.5	746	9.5	644	665	696	16.4	**422**	439.2	458	9.7
Late01	4750	4891.4	5033	99.9	3922	4057.7	4279	100.5	**1355**	1481	1607	89.7
Late02	4795	4956.4	5148	119.4	3956	4211.8	4439	170.9	**1297**	1518.1	1704	121.1
Late03	4543	4780	5090	147.8	3878	4019	4231	116.9	**1288**	1521.3	1639	115.4
Late04	4783	4975	5084	104.7	3980	4115.8	4279	86.9	**1385**	1503.3	1664	91.1
Late05	3906	4053.5	4209	101.3	3286	3435.3	3607	91.9	**993**	1164	1337	106.4
Hidden01	5111	5246.8	5416	90.3	4341	4418.3	4520	63.6	**1590**	1651.8	1728	51.1
Hidden02	1057	1076.4	1097	14.4	914	930.1	950	10.4	**401**	443.6	484	28.3
Hidden03	1001	1025.5	1046	16.6	827	871.5	941	31.4	**274**	330.4	367	32.7
Hidden04	930	970.8	995	17.4	813	848.7	901	27.1	**310**	336.7	368	19
Hidden05	1132	1166.8	1232	29.4	932	979.8	1046	39.1	**296**	362	431	35.1
Hint01	973	995.2	1025	18.7	843	870.4	899	21.3	**338**	368.8	421	25.2
Hint02	723	738.3	760	10.5	605	629.6	648	12.8	**235**	262.7	284	13
Hint03	3684	3839.3	3946	96.7	3031	3249.1	3377	89.8	**1040**	1173.9	1413	107.7

Table 18 Performance of PAR parameter settings for *sprint track* dataset.

Dataset	PAR = 0				PAR = 0.1				PAR = 0.4				PAR = 0.7			
	B.	M.	W.	Std.	B.	M.	W.	Std.	B.	M.	W.	Std.	B.	M.	W.	Std.
Early01	83	92	105	6.4	61	64.1	69	2.4	**60**	64.2	67	2.5	**60**	62.5	67	2.2
Early02	86	93.6	104	6.4	64	67.6	76	3.6	63	67	70	2.5	**62**	65.3	69	1.9
Early03	76	89.7	100	7.5	57	61.2	65	2.5	59	61	63	1.4	**53**	58.6	61	2.4
Early04	85	99.8	115	8.6	68	71.1	77	3.1	**64**	68.5	73	3.3	67	69.8	71	1.6
Early05	87	94.2	106	6.8	63	65.5	69	1.8	61	63.1	65	1.3	**60**	63.2	69	2.8
Early06	83	89.7	98	4.5	58	62.6	69	3.7	59	60.6	62	1	**57**	61.1	66	2.7
Early07	82	92	101	6.2	63	65.2	67	1.5	**59**	64.5	69	2.8	61	64.7	68	1.9
Early08	78	91.4	104	8.2	59	63.7	68	2.8	59	63.1	66	2.2	**58**	61.6	64	2.2
Early09	86	93.9	107	7.4	**61**	66.2	69	2.6	62	64.1	69	2.2	**61**	63.3	67	1.9
Early10	83	92.9	106	8.9	58	60.5	63	1.6	**56**	59.7	63	1.9	57	59.3	61	1.3
Late01	95	106.7	121	8.5	53	57.6	64	3.8	53	55.5	60	2.2	**52**	55.8	59	2.8
Late02	89	94.9	105	5.9	57	61.5	69	3.9	56	60.2	70	4.1	**54**	58.4	63	2.8
Late03	104	114.6	126	7.0	63	69.9	75	4.3	60	64.9	72	3.4	**59**	63.4	69	3.2
Late04	284	340	392	32.3	112	125.4	152	12	114	121.6	127	3.7	**104**	118.3	129	7.6
Late05	96	110.7	123	9.1	**55**	62.3	72	5	56	61.5	65	3.1	59	61.4	67	2.9
Late06	79	90.7	105	8.0	51	55.6	60	3.3	**50**	53.5	58	2.3	52	54.7	62	2.9
Late07	166	200.2	308	42.3	**60**	74.1	86	8.4	64	73.8	84	6.2	64	71.4	84	7.4
Late08	89	175.1	252	48.7	21	32.8	40	6.4	27	36.6	43	4.5	**17**	35.3	47	9
Late09	135	192.4	253	39.0	28	40.9	60	10.3	23	36	56	10.6	**17**	28.4	38	6.7
Late10	132	187	264	38.3	**61**	80.2	96	11.1	64	73.5	86	7.2	64	74.9	85	6.8
Hidden01	95	112.7	127	11.1	48	54	65	4.9	**43**	51.9	60	5.3	51	55.7	61	3.2
Hidden02	94	100.6	111	5.6	47	53.9	58	4.3	**45**	52	59	4.4	51	55.1	60	3.2
Hidden03	135	148	166	9.6	78	88.1	99	6.5	**75**	85	94	6.8	78	86	98	5.8
Hidden04	134	142.8	152	6.2	80	86.2	90	3.7	**79**	83.7	90	3.3	81	85.1	98	4.8
Hidden05	132	146.4	168	12.5	73	80.8	88	5	**72**	79.9	85	4.5	74	78.8	86	4.3
Hidden06	429	563.6	671	83.8	207	230	280	20.4	**202**	237.1	271	21.9	208	236	260	17.5
Hidden07	429	523.1	657	59.4	**196**	262.5	307	33.8	240	274.7	300	22.1	211	244.6	276	25.1
Hidden08	605	638.9	743	43.6	267	294.3	327	19.8	268	294.1	322	16.7	**266**	292	337	21.1
Hidden09	590	708.1	801	80.7	**373**	412.7	442	24.4	401	431.1	453	18.4	395	417.9	457	17.3
Hidden10	671	773.2	906	85.0	**346**	412.3	467	40.1	355	394.5	433	25.6	368	411.4	466	29.2
Hint01	221	294.2	344	40.1	**101**	120.4	136	10.7	102	116.6	130	8.7	103	112.4	126	8.1
Hint02	124	229.9	312	60.2	**59**	75.6	94	11.1	60	72	80	6.3	64	73.6	87	7
Hint03	209	307.2	358	51.9	84	97.6	108	8.2	80	102.5	117	11	**77**	100.8	119	12.2

Table 19 Performance of PAR parameter settings for *medium track* dataset.

Dataset	PAR = 0				PAR = 0.1				PAR = 0.4				PAR = 0.7			
	B.	M.	W.	Std.	B.	M.	W.	Std.	B.	M.	W.	Std.	B.	M.	W.	Std.
Early01	501	518.9	540	12.6	328	338.5	354	8.2	284	291.7	299	6	**270**	281.9	290	6.7
Early02	501	519.8	534	9.6	321	334.8	352	9	282	291.8	302	5.9	**275**	280.4	286	3.6
Early03	485	501.7	518	9.5	318	329.9	344	8.5	275	284.7	293	5.2	**265**	273.7	291	7.3
Early04	506	519.5	536	9.2	314	332.1	342	8.9	278	289.5	301	7.2	**263**	280	287	7.6
Early05	566	580.8	600	10.8	386	399.9	412	8.8	341	357.2	374	9.2	**334**	342.6	351	5.9
Late01	1535	1697.1	1855	105.7	366	411.8	461	35.9	276	293	317	12.5	**254**	282.6	297	13.2
Late02	410	440.3	494	23.0	104	126.3	145	12.6	89	92	97	2.3	**72**	79.8	89	7
Late03	438	489.3	528	29.9	116	141.6	153	11.2	78	89.3	98	7	**75**	84.7	99	8
Late04	413	438.1	474	17.7	103	121.6	132	10	91	97.2	104	4.8	**79**	87.1	97	6.9
Late05	1667	1780.7	1890	78.7	391	421.9	471	29	270	294.6	335	25.1	**238**	265.2	284	15.7
Hidden01	2120	2354.5	2607	156.7	460	514.1	615	47.2	279	309.4	334	15.3	**253**	283.3	298	13.1
Hidden02	1796	2017.4	2204	128.2	551	604.2	659	38.8	415	433.1	449	11.6	**361**	416.5	445	26.2
Hidden03	529	592.5	645	43.0	158	171.7	188	11.6	100	113.4	131	10.8	**93**	104	118	8.1
Hidden04	583	626.4	656	23.6	208	224.7	257	13.9	146	159	181	10.7	**135**	144.9	153	7
Hidden05	2049	2214.3	2365	118.0	455	502.8	555	37.1	**275**	342.6	396	37.4	280	323.5	367	28.6
Hint01	557	582.9	603	14.2	134	158.5	183	15.8	99	112.5	138	11.6	**89**	94.9	104	4.9
Hint02	1703	1835.5	1995	102.6	297	357	404	34.2	210	256.9	302	28.7	**194**	216.9	242	17.7
Hint03	2958	3529.3	3993	280.4	315	445	536	63.2	252	287.1	317	22.2	**242**	269.6	299	19.2

Table 20 Performance of PAR parameter settings for *long track* dataset.

Dataset	PAR = 0				PAR = 0.1				PAR = 0.4				PAR = 0.7			
	B.	M.	W.	Std.	B.	M.	W.	Std.	B.	M.	W.	Std.	B.	M.	W.	Std.
Early01	592	611.7	633	15.2	350	362.2	377	9.7	282	294.8	312	9.9	**256**	273.4	287	10.7
Early02	632	656.7	688	17.5	381	412.3	435	16.5	310	328.2	346	11.7	**299**	311	321	8
Early03	592	618.4	656	18.2	356	372.2	394	12.5	293	308.5	324	9.8	**286**	290.1	296	3.4
Early04	693	718	746	18.1	454	466.3	484	8.9	384	390.8	405	7.5	**356**	369.7	393	10.6
Early05	665	692.6	720	16.0	422	439.2	458	9.7	364	372.4	390	7.5	**337**	349.9	362	7
Late01	4035	4290.7	4493	140.8	1355	1481	1607	89.7	755	829.1	1048	83.9	**601**	673.8	789	63.1
Late02	4000	4297.2	4552	164.6	1297	1518.1	1704	121.1	741	837.3	970	69.2	**596**	669.6	718	39.9
Late03	3944	4154.5	4442	159.4	1288	1521.3	1639	115.4	717	819.1	960	72	**585**	670.6	745	50.9
Late04	4027	4373.1	4720	198.1	1385	1503.3	1664	91.1	801	920.9	1074	82.7	**621**	691.8	779	45
Late05	3196	3481.8	3780	162.0	993	1164	1337	106.4	555	644.2	732	56.6	**393**	491	541	45.9
Hidden01	4212	4527.9	4767	164.1	1590	1651.8	1728	51.1	901	989.4	1106	70.2	**747**	798.9	960	64.4
Hidden02	907	975.8	1021	39.2	401	443.6	484	28.3	240	272.7	296	18.2	**225**	241.8	279	14.9
Hidden03	819	897.5	935	32.9	274	330.4	367	32.7	151	170	185	10.2	**121**	130.9	141	5.9
Hidden04	814	879.3	953	38.5	310	336.7	368	19	169	184	203	12.7	**134**	147	162	9.6
Hidden05	916	999.3	1066	42.4	296	362	431	35.1	198	208.3	218	6.4	**146**	167.7	194	12.6
Hint01	843	880.3	926	22.0	338	368.8	421	25.2	192	215	246	17.9	**134**	163	191	16.1
Hint02	611	641.5	680	25.0	235	262.7	284	13	132	157	178	12.9	**102**	126.7	152	21
Hint03	3112	3387.9	3689	187.8	1040	1173.9	1413	107.7	501	585.2	655	55.6	**375**	494.2	579	59.5

Table 21 The performance of Global-best selection for *sprint track* dataset.

Dataset	Global-best selection				Random selection			
	B.	M.	W.	Std.	B.	M.	W.	Std.
Early01	60	62.5	67	2.2	**58**	59.4	61	1.1
Early02	62	65.3	69	1.9	**60**	62.4	65	1.8
Early03	**53**	58.6	61	2.4	**53**	56	58	1.7
Early04	67	69.8	71	1.6	**62**	67.8	88	8.4
Early05	60	63.2	69	2.8	**59**	62.1	75	4.7
Early06	57	61.1	66	2.7	**56**	56.8	58	0.8
Early07	61	64.7	68	1.9	**58**	62.5	75	5.3
Early08	58	61.6	64	2.2	**57**	58.5	60	1.1
Early09	61	63.3	67	1.9	**57**	62.5	74	5.9
Early10	57	59.3	61	1.3	**53**	56.1	58	1.7
Late01	52	55.8	59	2.8	**45**	58.4	99	20.1
Late02	54	58.4	63	2.8	**49**	62.7	85	14.8
Late03	59	63.4	69	3.2	**56**	76.9	115	21.7
Late04	**104**	118.3	129	7.6	279	346.4	440	56.1
Late05	59	61.4	67	2.9	**51**	57.7	80	10.6
Late06	52	54.7	62	2.9	**43**	48.2	57	4
Late07	**64**	71.4	84	7.4	68	117.9	172	33.8
Late08	**17**	35.3	47	9	22	73.3	128	39.4
Late09	**17**	28.4	38	6.7	86	125.4	177	24.3
Late10	**64**	74.9	85	6.8	130	156.7	183	17.8
Hidden01	51	55.7	61	3.2	**41**	87.3	118	31.6
Hidden02	51	55.1	60	3.2	**35**	40.8	45	4.2
Hidden03	78	86	98	5.8	**70**	92.5	138	26.5
Hidden04	**81**	85.1	98	4.8	138	156.8	167	10.5
Hidden05	74	78.8	86	4.3	**62**	83.2	123	23.2
Hidden06	**208**	236	260	17.5	486	687	819	101.3
Hidden07	**211**	244.6	276	25.1	286	436.3	637	108.7
Hidden08	**266**	292	337	21.1	490	644.4	843	107.6
Hidden09	**395**	417.9	457	17.3	874	945.2	976	33.9
Hidden10	**368**	411.4	466	29.2	599	663.3	780	62.9
Hint01	**103**	112.4	126	8.1	211	317.1	399	69.2
Hint02	64	73.6	87	7	**62**	150	239	59.1
Hint03	**77**	100.8	119	12.2	192	294.2	436	72.4

Table 22 The performance of Global-best selection for *medium track* dataset.

Dataset	Global-best selection				Random selection			
	B.	M.	W.	Std.	B.	M.	W.	Std.
Early01	**270**	281.9	290	6.7	443	450.9	457	5.2
Early02	**275**	280.4	286	3.6	434	447.8	460	9.2
Early03	**265**	273.7	291	7.3	431	440.5	447	4.5
Early04	**263**	280	287	7.6	440	448.3	457	5.5
Early05	**334**	342.6	351	5.9	501	511.9	520	6.3
Late01	**254**	282.6	297	13.2	1758	1802.5	1855	30
Late02	**72**	79.8	89	7	412	425.9	440	8.4
Late03	**75**	84.7	99	8	477	507.8	536	16
Late04	**79**	87.1	97	6.9	405	435.2	465	15.8
Late05	**238**	265.2	284	15.7	1746	1832.7	1922	47.7
Hidden01	**253**	283.3	298	13.1	2440	2567.1	2688	76
Hidden02	**361**	416.5	445	26.2	2069	2127.7	2174	32.8
Hidden03	**93**	104	118	8.1	628	641.2	656	10.9
Hidden04	**135**	144.9	153	7	615	631.4	646	11.8
Hidden05	**280**	323.5	367	28.6	2313	2372.9	2466	44.3
Hint01	**89**	94.9	104	4.9	575	595.9	624	14.2
Hint02	**194**	216.9	242	17.7	1910	2043.9	2145	70.4
Hint03	**242**	269.6	299	19.2	3912	4190.1	4420	175.3

petitive results in comparison with those obtained by the winners' methods in the remaining datasets. The symbol '√' indicates that the winner method obtained the best result while the symbol '–' denotes its inability to do so.

6. Conclusion

This major contribution of this paper is an improvement made to the Harmony Search Algorithm (HSA) for the Nurse Rostering Problem (NRP). Nurse Rostering as a real-world

Table 23 The performance of Global-best selection for *long track* dataset.

Dataset	Global-best selection				Random selection			
	B.	M.	W.	Std.	B.	M.	W.	Std.
Early01	**256**	273.4	287	10.7	492	511.9	526	11.9
Early02	**299**	311	321	8	550	561.8	576	8.3
Early03	**286**	290.1	296	3.4	489	503.8	513	8.4
Early04	**356**	369.7	393	10.6	587	611	623	11.6
Early05	**337**	349.9	362	7	561	581.9	596	11.9
Late01	**601**	673.8	789	63.1	3734	3825.6	4004	82
Late02	**596**	669.6	718	39.9	3712	3883.2	3980	89.2
Late03	**585**	670.6	745	50.9	3537	3671.1	3754	62.5
Late04	**621**	691.8	779	45	3583	3752.8	3895	87.1
Late05	**393**	491	541	45.9	3058	3188	3292	63.6
Hidden01	**747**	798.9	960	64.4	3976	4128.2	4236	78.9
Hidden02	**225**	241.8	279	14.9	818	846	872	18.3
Hidden03	**121**	130.9	141	5.9	766	785.3	810	16.9
Hidden04	**134**	147	162	9.6	719	738.2	759	12.1
Hidden05	**146**	167.7	194	12.6	881	920.5	999	34.4
Hint01	**134**	163	191	16.1	760	782.8	806	15.5
Hint02	**102**	126.7	152	21	559	584.6	594	9.8
Hint03	**375**	494.2	579	59.5	3036	3105.4	3160	44.3

Table 24 INRC2010 winners' methods.

Key	Method	Reference
M_1	*Hyper-heuristic combined with a greedy shuffle approach.*	Bilgin et al. (2010)
M_2	*Variable Depth Search Algorithm and Branch and Price Algorithm.*	Burke and Curtois (2010)
M_3	*Tabu search with restart mechanism.*	Lu and Hao (2010)
M_4	*Constraint Optimization Problem solver.*	Nonobe (2010)
M_5	*Integer programming with set of neighborhood structures.*	Valouxis et al. (2010))

optimization problem is considered to be NP-hard, which is not easy to solve. HSA is able to solve the NRP efficiently like the other real-world problems solved by HSA: water distribution networks, course timetabling, examination timetabling, etc. The HSA for NRP has been improved in two aspects: first, the Global-best selection of Particle Swarm Optimization replaced the random selection in memory consideration during the improvisation process to improve the convergence speed. Second, multi-pitch adjustment procedures have been established to improve local exploitation capability. The results obtained by the proposed method are positively comparable with those provided by the five winners' methods in INRC2010.

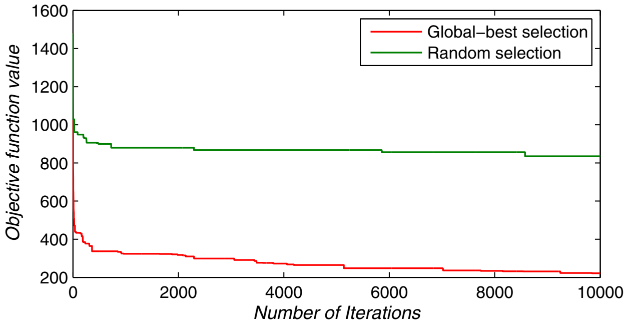

Figure 1 The results distribution in HM of different selection methods using long_hidden03 dataset.

The effectiveness of the two proposed improvements to the HSA for NRP has been carried out using eight experimental cases, each with a different parameter setting. Experimentally, for the first improvement, the Global-best selection combined with the process of memory consideration has been able to improve the results considerably. This proves that the Global-

Table 25 A comparison between the results of HSA and winners' methods for *sprint track* dataset.

Datasets	Proposed HSA	MHSA	Competitive methods					
			Best result	M_1	M_2	M_3	M_4	M_5
Early01	58	60	**56**	–	√	√	√	√
Early02	60	61	**58**	–	√	√	√	√
Early03	53	56	**51**	–	√	√	√	√
Early04	62	66	**59**	√	√	–	√	√
Early05	59	61	**58**	√	√	√	√	√
Early06	56	58	**54**	√	√	√	√	√
Early07	58	62	**56**	√	√	√	√	√
Early08	57	59	**56**	–	√	√	√	√
Early09	57	57	**55**	–	√	√	√	√
Early10	53	58	**52**	–	√	√	√	√
Late01	45	47	**37**	–	√	–	–	√
Late02	49	53	**42**	–	√	–	–	√
Late03	55	59	**48**	–	√	–	√	√
Late04	104	117	**75**	–	√	–	–	–
Late05	51	54	**44**	–	√	–	–	√
Late06	43	47	**42**	–	√	√	√	√
Late07	60	66	**42**	–	√	–	–	–
Late08	**17**	19	**17**	–	√	√	√	√
Late09	**17**	34	**17**	–	√	√	√	√
Late10	54	73	**43**	–	√	–	–	–
Hidden01	41	48	**33**	–	–	–	√	√
Hidden02	35	45	**32**	√	–	–	√	–
Hidden03	70	76	**62**	–	–	–	√	√
Hidden04	79	97	**67**	–	–	–	√	√
Hidden05	62	68	**59**	√	–	–	–	–
Hidden06	202	278	**134**	–	–	–	√	–
Hidden07	196	201	**153**	–	–	–	–	√
Hidden08	266	374	**209**	–	–	–	√	–
Hidden09	373	916	**338**	–	–	–	–	√
Hidden10	346	462	**306**	–	–	–	–	√
Hint01	101	104	**78**	–	√	–	–	–
Hint02	59	73	**47**	–	√	–	–	–
Hint03	77	92	**57**	–	√	–	–	–

Table 26 A comparison between the results of HSA and winners' methods for *medium track* dataset.

Datasets	Proposed HSA	Competitive methods					
		Best result	M_1	M_2	M_3	M_4	M_5
Early01	270	**240**	–	√	√	–	√
Early02	275	**240**	–	√	–	√	√
Early03	265	**236**	–	√	–	√	√
Early04	263	**237**	–	√	–	–	√
Early05	334	**303**	–	√	–	–	√
Late01	254	**158**	–	√	–	–	–
Late02	72	**18**	–	√	–	–	–
Late03	75	**29**	–	√	–	–	–
Late04	79	**35**	–	√	–	–	–
Late05	238	**107**	–	√	–	–	–
Hidden01	253	**130**	–	–	–	√	–
Hidden02	361	**221**	–	–	–	–	√
Hidden03	93	**36**	–	–	–	√	–
Hidden04	135	**80**	–	–	–	–	√
Hidden05	275	**122**	–	–	–	–	√
Hint01	89	**40**	√	–	–	–	–
Hint02	194	**84**	√	–	–	–	–
Hint03	242	**129**	√	–	–	–	–

Table 27 A comparison between the results of HSA and winners' methods for *long track* dataset.

Datasets	Proposed HSA	Competitive methods					
		Best result	M_1	M_2	M_3	M_4	M_5
Early01	256	**197**	√	√	–	√	√
Early02	299	**219**	–	√	–	–	√
Early03	286	**240**	√	√	–	√	√
Early04	356	**303**	√	√	–	√	√
Early05	337	**284**	√	√	–	√	√
Late01	601	**235**	–	√	–	–	–
Late02	596	**229**	–	√	–	–	–
Late03	585	**220**	–	√	–	–	–
Late04	621	**221**	–	√	–	–	–
Late05	393	**83**	–	√	–	–	√
Hidden01	747	**363**	–	–	–	–	√
Hidden02	225	**90**	√	–	–	–	–
Hidden03	121	**38**	–	–	–	–	√
Hidden04	134	**22**	–	–	–	–	√
Hidden05	146	**41**	–	–	–	–	√
Hint01	134	**31**	√	–	–	–	–
Hint02	102	**17**	√	–	–	–	–
Hint03	375	**53**	√	–	–	–	–

best has a direct effect on the convergence of HSA. For the second improvement, the multi-pitch adjustment procedures with larger PAR have been able to empower the search to greatly exploit the NRP search space and thus have improved the local nearby exploitation.

It would be interesting if other researchers can explore the following:

1. the unfeasible regions, as our paper was concerned with exploring the feasible ones;
2. other local changes in the pitch adjustment operator;
3. integration of HSA with other approximation-based methods to improve the HSA performance.

References

Abual-rub, M.S., Al-betar, M.A., Abdullah, R., Khader, A.T., 2012. A hybrid harmony search algorithm for ab initio protein tertiary structure prediction. Network Modeling and Analysis in Health Informatics and Bioinformatics, 1–17. http://dx.doi.org/10.1007/s13721-012-0013-7.

Aickelin, U., Dowsland, K.A., 2004. An indirect genetic algorithm for a nurse-scheduling problem. Computers & Operations Research 31 (5), 761–778.

Al-betar, M., Khader, A., Liao, I., 2010a. A harmony search with multi-pitch adjusting rate for the university course timetabling. Recent Advances In Harmony Search Algorithm, 147–161.

Al-betar, M.A., Doush, I.A., Khader, A.T., Awadallah, M.A., 2012a. Novel selection schemes for harmony search. Applied Mathematics and Computation 218, 6095–6117.

Al-betar, M.A., Khader, A.A, 2009. Hybrid harmony search for university course timetabling. In: Proceedings of the 4nd Multidisciplinary Conference on Scheduling: Theory and Applications (MISTA 2009), Dublin, Ireland, August 2009, pp. 157–179.

Al-betar, M.A., Khader, A.T., Nadi, F., 2010b. Selection mechanisms in memory consideration for examination timetabling with harmony search. In: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation. ACM, pp. 1203–1210.

Al-betar, M.A., Khader, A.T., Thomas, J.J.A, 2010. Combination of metaheuristic components based on Harmony Search for the uncapacitated examination timetabling. In: 8th International Conference on the Practice and Theory of Automated Timetabling (PATAT 2010), Belfast, Northern Ireland, August 2010, pp. 57–80.

Al-betar, M.A., Khader, A.T., Zaman, M., 2012b. University course timetabling using a hybrid harmony search metaheuristic algorithm. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 99, 1–18. http://dx.doi.org/10.1109/TSMCC.2011.2174356.

Alia, O.M., Mandava, R., 2011. The variants of the harmony search algorithm: an overview. Artificial Intelligence Review 36 (1), 49–68.

Awadallah, M., Khader, A., Al-betar, M., Bolaji, A., 2011a. In: Panigrahi, B., Suganthan, P., Das, S., Satapathy, S. (Eds.), Nurse Rostering Using Modified Harmony Search Algorithm, SEMCCO'11, vol. 7077. LNCS, pp. 27–37.

Awadallah, M.A., Khader, A.T., Al-betar, M.A., Bolaji, A.L., 2011. Nurse scheduling using harmony search. Bio-inspired computing: theories and applications (BIC-TA). In: 2011 Sixth International Conference, IEEE, pp. 58–63.

Bartholdi III, J.J., 1981. A guaranteed-accuracy round-off algorithm for cyclic scheduling and set covering. Operations Research, 501–510.

Bilgin, B., De causmaecker, P., Rossie, B., Vanden berghe, G., 2011. Local search neighbourhoods for dealing with a novel nurse rostering model. Annals of Operations Research 194 (1), 33–57.

Bilgin, B., Demeester, P., Misir, M., Vancroonenburg, W., Vanden berghe, G., Wauters, T., 2010. A Hyper-Heuristic Combined with a Greedy Shuffle Approach to the Nurse Rostering Competition. INRC2010 (http://www.kuleuven-kortrijk.be/nrpcompetition).

Blum, C., Roli, A., 2003. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. ACM Computing Surveys (CSUR) 35 (3), 268–308.

Brusco, M.J., Jacobs, L.W., 1995. Cost analysis of alternative formulations for personnel scheduling in continuously operating organizations. European Journal of Operational Research 86 (2), 249–261.

Burke, E.K., Curtois, T., 2010. An Ejection Chain Method and a Branch and Price Algorithm Applied to the Instances of the First International Nurse Rostering Competition. INRC2010 (http://www.kuleuven-kortrijk.be/nrpcompetition).

Burke, E.K., Curtois, T., Post, G., Qu, R., Veltman, B., 2008. A hybrid heuristic ordering and variable neighbourhood search for the nurse rostering problem. European Journal of Operational Research 188 (2), 330–341.

Burke, E.K., Curtois, T., Qu, R., Berghe, G.V., 2009. A scatter search methodology for the nurse rostering problem. Journal of the Operational Research Society 61 (11), 1667–1679.

Burke, E.K., De causmaecker, P., Berghe, G.V., Van landeghem, H., 2004. The state of the art of nurse rostering. Journal of Scheduling 7 (6), 441–499.

Burke, E.K., De causmaecker, P., Vanden berghe, G., 1999. A hybrid tabu search algorithm for the nurse rostering problem. Simulated Evolution and Learning (1585/1999), 187–194.

Burke, E.K., Li, J., Qu, R., 2010. 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.

Cheang, B., Li, H., Lim, A., Rodrigues, B., 2003. Nurse rostering problems – a bibliographic survey. European Journal of Operational Research 151 (3), 447–460.

Dowsland, K.A., 1998. Nurse scheduling with tabu search and strategic oscillation. European Journal of Operational Research 106 (2–3), 393–407.

Forsati, R., Haghighat, A., Mahdavi, M., 2008. Harmony search based algorithms for bandwidth-delay-constrained least-cost multicast routing. Computer Communications 31 (10), 2505–2519.

Geem, Z.W., 2006. Optimal cost design of water distribution networks using harmony search. Engineering Optimization 38, 259–277.

Geem, Z.W., 2008. Novel derivative of harmony search algorithm for discrete design variables. Applied Mathematics and Computation 199 (1), 223–230.

Geem, Z.W., Kim, J.H., Loganathan, G.V., 2001. A new heuristic optimization algorithm: harmony search. Simulation 76 (2), 60–68.

Gutjahr, W.J., Rauner, M.S., 2007. An ACO algorithm for a dynamic regional nurse-scheduling problem in Austria. Computers & Operations Research 34 (3), 642–666.

Ingram, G., Zhang, T., 2009. Overview of applications and developments in the harmony search algorithm. Music-Inspired Harmony Search Algorithm (191/2009), 15–37.

Lee, K.S., Geem, Z.W., 2005. 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.

Lee, K.S., Geem, Z.W., Lee, S., Bae, K., 2005. The harmony search heuristic algorithm for discrete structural optimization. Engineering Optimization 37 (7), 663–684.

Lu, Z., Hao, J.-K., 2010. Adaptive Local Search for the First International Nurse Rostering Competition. INRC2010 (http://www.kuleuven-kortrijk.be/nrpcompetition).

Maenhout, B., Vanhoucke, M., 2007. An electromagnetic meta-heuristic for the nurse scheduling problem. Journal of Heuristics 13 (4), 359–385.

Maenhout, B., Vanhoucke, M., 2010. Branching strategies in a branch-and-price approach for a multiple objective nurse scheduling problem. Journal of Scheduling 13 (1), 77–93.

Mahdavi, M., Fesanghary, M., Damangir, E., 2007. An improved harmony search algorithm for solving optimization problems. Applied Mathematics and Computation 188, 1567–1579.

Millar, H.H., Kiragu, M., 1998. Cyclic and non-cyclic scheduling of 12 h shift nurses by network programming. European Journal of Operational Research 104 (3), 582–592.

Nonobe, K., 2010. INRC2010: An Approach Using a General Constraint Optimization Solver. INRC2010 (http://www.kuleuven-kortrijk.be/nrpcompetition).

Omran, M.G.H., Mahdavi, M., 2008. Global-best harmony search. Applied Mathematics and Computation 198, 643–656.

Rizzato, D., Constantino, A., Luiz de melo, E., Landa-silva, D., Romao, W., 2010. Heuristic Algorithm Based On Multi-Assignment Problems For Nurse Rostering Problem. INRC2010 (http://www.kuleuven-kortrijk.be/nrpcompetition).

Sivasubramani, S., Swarup, K., 2011. Multi-objective harmony search algorithm for optimal power flow problem. International Journal of Electrical Power & Energy Systems 33 (3), 745–752.

Tsai, C.C., Li, S.H.A., 2009. A two-stage modeling with genetic algorithms for the nurse scheduling problem. Expert Systems with Applications 36 (5), 9506–9512.

Valouxis, C., Gogos, C., Goulas, G., Alefragis, P., Housos, E., 2010. A systematic two phase approach for the Nurse Rostering Problem. INRC2010 (http://www.kuleuven-kortrijk.be/nrpcompetition).

Wang, L., Pan, Q.K., Fatih tasgetiren, M., 2010. Minimizing the total flow time in a flow shop with blocking by using hybrid harmony search algorithms. Expert Systems with Applications 37 (12), 7929–7936.