HARMONY SEARCH WITH NOVEL SELECTION METHODS IN MEMORY CONSIDERATION FOR NURSE ROSTERING PROBLEM

MOHAMMED A. AWADALLAH*
School of Computer Sciences
Universiti Sains Malaysia (USM)
11800 Pulau Pinang, Malaysia
mama10_com018@student.usm.my

AHAMAD TAJUDIN KHADER
School of Computer Sciences
Universiti Sains Malaysia (USM)
11800 Pulau Pinang, Malaysia
tajudin@cs.usm.my

MOHAMMED AZMI AL-BETAR
Department of Computer Science
Al-Huson University College
Al-Balqa Applied University
P.O. Box 50, Al-Huson, Irbid, Jordan
School of Computer Sciences
Universiti Sains Malaysia (USM)
11800 Pulau Pinang, Malaysia
mohbetar@cs.usm.my

ASAJO LA’ARO BOLAJI
School of Computer Sciences
Universiti Sains Malaysia (USM)
11800 Pulau Pinang, Malaysia
Department of Computer Science
University of Ilorin
Ilorin, Nigeria
abl10_sa0739@student.usm.my

Received 4 July 2012
Revised 19 April 2013
Accepted 10 September 2013
Published 16 December 2013

The selection methods of population-based metaheuristics provide the driving force to generate good solutions. These selection methods select the individuals with a

*Corresponding author
higher fitness to be members of the population in the next iteration correspond to the natural rule of Darwin’s principle survival-of-the-fittest. Harmony search algorithm is a population-based metaheuristic, which mimicking the musical improvisation process where a group of musicians play the pitches of their musical instruments seeking for a pleasing harmony. It improvises the new harmony based on three rules: memory consideration, random consideration, and pitch adjustment. In this paper, we investigate the replacement of the original random selection of memory consideration with a set of selection methods in order to speed-up the convergence. These selection methods include tournament, proportional, and linear rank of Genetic Algorithm, and Global-best of Particle Swarm Optimization. The proposed harmony search with the different memory consideration selection methods evaluated using standard dataset published in the first International Nurse Rostering Competition INRC2010. Nurse rostering problem is a combinatorial optimization problem tackled by assigning a set of nurses with different skills to a set of shifts over predefined scheduling period. Experimentally, the tournament memory consideration selection method achieved the best rate of convergence as well as the best results in comparison with the other memory consideration selection methods.

Keywords: Harmony search; selection methods; metaheuristics; population-based; nurse rostering.

1. Introduction

Nurse Rostering Problem (NRP) is a combinatorial optimization problem (Millar and Kiragu, 1998), tackled by assigning a set of nurses to a set of different shifts over a predefined scheduling period. 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 the 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. The basic objective is to achieve a feasible roster with a good quality.

NRP has attracted many researchers over time since 1960 and there are many methods investigated to tackle this problem. These methods include Genetic Algorithm (GA) (Cai and Li, 2000; Tsai and Li, 2009), Ant Colony Optimization (Gutjahr and Rauner, 2007), Tabu Search (Dowsland, 1998; Burke et al., 1999), Simulated Annealing (Brusco and Jacobs, 1995), Variable Neighborhood Search (Burke et al., 2008; Bilgin et al., 2012), and Scatter Search (Burke et al., 2009). For more details, see the surveys of (Cheang et al., 2003; Burke et al., 2004; den Bergh Jorne et al., 2013). It should be noted that all of these methods could not obtained exact solutions for any of the problem instances due to their complexity and the contradictory between constraints. Clearly, the research in the field of NRP is still active and presenting a new potential method like HSA could benefit the domain with a new optimization template that might be able to achieve promising results.

Optimization is a process that aims to select the best values for different problem variables under certain constraints (if any). Harmony Search Algorithm (HSA) is a one of the latest population-based metaheuristic methods proposed by Geem et al. (2001). It has been used for solving different optimization problems such as: The blocking permutation flow shop scheduling problem (Wang et al., 2011), the optimal power flow problem (Sivasubramani and Swarup, 2011), the
multicast routing problem (Forsati et al., 2008), water pump switching (Geem, 2005), course timetabling (Al-Betar and Khader, 2009; Al-Betar et al., 2012), examination timetabling (Al-Betar et al., 2010a; 2010b), office space allocation (Awadallah et al., 2012), and many others reported in (Ingram and Zhang, 2009; Alia and Mandava, 2011; Manjarres et al., 2013). HSA has attracted the attention of several researchers to experiment with it due to its impressive characteristics: (i) it has novel stochastic derivative criteria (Geem, 2008), (ii) it does not require initial values for the decision variables, (iii) it iteratively generates a new solution by considering all existing solutions in the population, and (iv) it is used for continuous and discrete problems (Lee and Geem, 2005). The performance of HSA is improved by tuning its parameters (Mahdavi et al., 2007; Alatas, 2010; Geem and Sim, 2010) or by hybridizing it with other methods (Fesanghary et al., 2008; Omran and Mahdavi, 2008; Kaveh and Talatahari, 2009; Zou et al., 2011; Al-Betar et al., 2012).

HSA mimics the musical improvisation process in which a group of musicians play the pitches of their musical instruments together seeking for a pleasing harmony as determined by audio-aesthetic standards. HSA starts with a population of solutions kept in harmony memory (HM). It improvises the new harmony iteratively using three operators: memory consideration that selects the variables of the new harmony from HM 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. At each iteration, a new harmony is generated and substituted with the worst solution in the HM, if better. This process is repeated until it converges.

There are many selection methods in population-based metaheuristics investigated to select the individuals during the recombination and mutation operators to generate good-enough solutions. These selection methods include random selection, tournament selection, proportional selection, etc. Naturally, the selection methods select the individuals with high fitness from the population to guide the search towards promising regions of the search space in the hope to obtain good solutions. The principle of these selection methods extract form the natural rule of Darwins “survival-of-the-fittest”. The HSA like other population-based methods on applying the different selection methods in pitch adjustment or memory consideration operators to overcome its weakness. In pitch adjustment operator, Omran and Mahdavi (2008) modified the pitch adjustment operator by adjust the values of the new harmony with the values of Global-best solution in HM, which is called Global-best Harmony Search (GHS). The performance of GHS is better than the classical HSA. In other studies, we investigated novel selection methods in memory consideration of HSA for continuous optimization problems in Al-Betar et al. (2011) and Al-Betar et al. (2013). These methods tested using standard benchmark mathematical functions. Experimentally, these methods led to improvement in the performance of the HSA. The authors concluded that the selection method on memory consideration has a direct impact on the performance of HSA.
Recently, Awadallah et al. (2011b) adapted HSA for NRP using INRC2010 dataset. Experimentally, this method achieved promising results especially for small instances of INRC2010 dataset. In another development, Awadallah et al. (2011a) modified the HSA by integrating specific local search procedures in the pitch adjustment operator in order to cater for the soft constraints violation. On the other hand, the modified HSA is hybridized with greedy shuffle local search procedure to locally enhance the new solution at each iteration (Awadallah et al., 2012). The three harmony search-based algorithms proposed in Awadallah et al. (2011a,b) and Awadallah et al. (2012) suffer from the slow rate of convergence especially for the hardest instances. This is because they couldn’t considered the natural rule of Darwin’s survival-of-the-fittest in memory consideration operator. Note that in the original memory consideration, the values of the decision variables are taken from any solution stored in HM, not from the fittest solutions.

In this paper, we investigated selection methods in the memory consideration of HSA for solving NRP. This is to study the possible effect of integrating these selection methods with the memory consideration on the performance of HSA using a discrete optimization problem like NRP. The main concern of these selection methods is to employ the survival-of-the-fittest principle and thus emphasizing on the most promising search space regions during the search. It is noteworthy that the selection methods used in this paper include tournament, proportional, linear rank of GA, and global-best of Particle Swarm Optimization (PSO). The proposed HSA variations (i.e. five HSA variations are proposed where each of which is HSA with one selection method in memory consideration) are tested using standard dataset introduced by the first International Nurse Rostering Competition INRC2010. The results of the experimental showed that the variation of HSA with tournament memory consideration selection (TMCS) method achieved best convergence rate as well as the best results in comparison with other variations of HSA.

The remainder of the paper is organized as follows: Sec. 2, we are overviews the NRP we were dealing with, where Sec. 3 provides an overview of procedural steps of HSA. Section 4 describes HSA with different memory consideration selection methods. The results of application of HSA with the different selection methods to NRP is given in Sec. 5. The conclusion and future directions are described in Sec. 6.

2. Problem Description

The NRP is tackled by assigning a set of nurses with various skills and work contracts, to a set of shift types over a predefined scheduling period. The NRP solution (or roster) is subject to hard and soft constraints. The hard constraints (see $H_1$, $H_2$ in Table 1) must be fulfilled in the roster to be feasible. The fulfillment of soft constraints (see $S_1$–$S_{10}$ in Table 1) is desirable to satisfy as much as possible, and used to determine the quality of the roster. The basic objective is to find a feasible roster with a good quality.
Table 1. INRC2010 hard and soft constraints.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>The constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H_1$</td>
<td>All demanded shifts must be assigned to a nurse.</td>
</tr>
<tr>
<td>$H_2$</td>
<td>A nurse can only work one shift per day, i.e. no two shifts can be assigned to the same nurse on a day.</td>
</tr>
<tr>
<td>$S_1$</td>
<td>Maximum and minimum number of assignments for each nurse during the scheduling period.</td>
</tr>
<tr>
<td>$S_2$</td>
<td>Maximum and minimum number of consecutive working days.</td>
</tr>
<tr>
<td>$S_3$</td>
<td>Maximum and minimum number of consecutive free days.</td>
</tr>
<tr>
<td>$S_4$</td>
<td>Assign complete weekends.</td>
</tr>
<tr>
<td>$S_5$</td>
<td>Assign identical complete weekends.</td>
</tr>
<tr>
<td>$S_6$</td>
<td>Two free days after a night shift.</td>
</tr>
<tr>
<td>$S_7$</td>
<td>Requested day-on/off.</td>
</tr>
<tr>
<td>$S_8$</td>
<td>Requested shift-on/off.</td>
</tr>
<tr>
<td>$S_9$</td>
<td>Alternative skill.</td>
</tr>
<tr>
<td>$S_{10}$</td>
<td>Unwanted patterns. (where pattern as a set of legal shifts defined in terms of work to be done during the shifts (Wren, 1996)).</td>
</tr>
</tbody>
</table>

Table 2. An example set of shift types and weekly nurses demand.

<table>
<thead>
<tr>
<th>Shift type</th>
<th>Weekly nurses demand</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mon</td>
</tr>
<tr>
<td>Day Shift (D)</td>
<td>7</td>
</tr>
<tr>
<td>Early Shift (E)</td>
<td>3</td>
</tr>
<tr>
<td>Late Shift (L)</td>
<td>5</td>
</tr>
<tr>
<td>Night Shift (N)</td>
<td>2</td>
</tr>
</tbody>
</table>

The hospital administration organizes work by dividing each day during the scheduling period to a set of shifts such as day, late, early, and night shifts. Each shift requires a predefined number of nurses (i.e. demand nurses) each with specific skills to fulfill the different services provided (see Table 2).

Each nurse can be assigned to a set of shifts based on his/her skills and details of work contract. The nurse with higher skill level can be assigned to the shifts with the same or lower skill level. The contract defines the job specification for the nurse that includes the maximum and minimum number of assignments during the scheduling period, the maximum and minimum number of consecutive working days, the maximum and minimum number of consecutive free days, the weekend days (such as Saturday-Sunday, or Friday-Saturday-Sunday), assignment of complete and identical weekend, assignment of two free days after night shift, defined alternative skills for the nurse, and determined patterns (i.e. consecutive group of shifts) that the nurse does not like to work on. The contract details are transformed to a soft constraints ($S_1$–$S_6$, $S_9$, and $S_{10}$ in Table 1).

The preferences of the nurse are collected before generating the roster, and these preferences are changed from scheduling period to another. These preferences...
include the day-off, day-on, shift-off, and shift-on which are part of the soft constraints ($S_7$ and $S_8$ in Table 1). Notably, the mathematical formulation for NRP using INRC2010 dataset presented in Awadallah et al. (2013).

The roster is evaluated using the objective function formalized in (1) that adds up the penalty of soft constraint violations in a feasible roster.

$$\min f(x) = \sum_{s=1}^{10} c_s \cdot g_s(x).$$

(1)

It should be noted 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 $g_s(x)$ is the total number of violations for the soft constraint $s$ in solution roster $x$.

3. 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 audio-aesthetic standards. Each musician improvises any pitch within the possible pitch range, together making one new harmony. If all the pitches make a good harmony according to the audio-aesthetic standard, that experience is kept in each musicians memory, and the possibility of making a good harmony is increased in the next practice. During each practice, when the musician improvises one pitch, he/she follows any of these three rules: (i) playing any of the pitches from HM; (ii) modifying a pitch which exists in his/her memory; and (iii) playing a random pitch within its possible range.

Analogically, in the optimization, a set of decision variables are assigned with values, iteration by iteration, seeking (near-) optimal solution as evaluated by an objective function. Each decision variable is assigned with value from the possible range of values, together making one solution vector. If all the values of decision variables make a good solution, that solution vector will be stored in memory and the possibility to construct a good solution is also increased in the next iteration. Note that, in each iteration, each decision variable is assigned one value depending on the following three rules: (i) choosing a random value from HM; (ii) choosing an adjacent value of one value stored in HM; and (iii) choosing a random value from the range of possible values. These rules were formulated by Geem et al. (2001) as operators of the HSA that include: (i) memory consideration, (ii) pitch adjustment, and (iii) random consideration respectively. HSA has five main steps will be described below, and the HSA pseudo-code is provided in Algorithm 1.

3.1. Initialize the parameters of optimization problem and HSA

In general, the optimization problems is modeled as follows:

$$\min f(x) \text{ subject to } x_j \in X_j, \text{ where } j \in [1, N],$$
Algorithm 1 The harmony search algorithm

STEP1 Initialize the parameters of the optimization problem and HSA
1: Input the dataset of the optimization problem.
2: Set the HSA parameters (HMCR, PAR, NI, HMS).

STEP2 Initialize harmony memory (HM)
1: Construct the solution vectors of the harmony memory, $HM = \{x^1, x^2, \ldots, x^{HMS}\}$
2: Identify the worst vector in $HM$, $x_{\text{worst}} \in \{x^1, x^2, \ldots, x^{HMS}\}$

STEP3 Improvise a new harmony
1: $x' = \phi$ \hspace{1cm} \{new harmony vector\}
2: \textbf{for} each $j \in [1, N]$ \textbf{do}
3: \hspace{1cm} if $(U(0, 1) \leq \text{HMCR})$ then
4: \hspace{2cm} $x'_j = x^i_j$, where $i \sim U(1, \cdots, HMS)$ \hspace{0.5cm} \{memory consideration\}
5: \hspace{2cm} \textbf{if} $(U(0, 1) \leq \text{PAR})$ then
6: \hspace{3cm} $x'_j = v_{j,k} \pm m$ \hspace{0.5cm} \{pitch adjustment\}
7: \hspace{2cm} \textbf{end if}
8: \hspace{1cm} \textbf{else}
9: \hspace{2cm} $x'_j \in X_j$ \hspace{0.5cm} \{random consideration\}
10: \hspace{1cm} \textbf{end if}
11: \textbf{end for}

STEP4 Update the harmony memory
1: \textbf{if} $(f(x') < f(x_{\text{worst}}))$ then
2: \hspace{1cm} Replaces $x_{\text{worst}}$ by $x'$ in the HM.
3: \textbf{end if}

STEP5 Check the stopping criterion
1: \textbf{while} (the maximum number of improvisations NI is not reached) \textbf{do}
2: \hspace{1cm} Repeat STEP3 and STEP4
3: \textbf{end while}

where $f(x)$ is the objective function; $x_j$ is the decision variable in the solution vector $x$; $X_j$ is the set of the possible range of values for the design variable $x_j$; and $N$ is the number of design variables in the solution vector $x$. The HSA parameters are also specified in this step. These are the harmony memory size (HMS), or the number of solution vectors in harmony memory; the harmony memory (HM); harmony memory consideration rate (HMCR), or the rate of selecting the values from HM vectors; pitch adjustment rate (PAR), or the rate of the local improvement; the number of improvisations (NI), or the number of iterations. It should be noted that the HMCR and PAR are parameters used in the improvisation process and they will be explained in more detail in the steps to come.
3.2. Initialize HM

The HM is a space of memory used to keep a set of solution vectors as the HMS, where these vectors randomly generated. Furthermore, the objective function value (or quality) for each vector is determined (see (2)).

\[
HM = \begin{bmatrix}
  x_1^1 & x_2^1 & \cdots & x_N^1 & f(x_1) \\
  x_1^2 & x_2^2 & \cdots & x_N^2 & f(x^2) \\
  \vdots & \vdots & \ddots & \vdots & \vdots \\
  x_1^{\text{HMS}} & x_2^{\text{HMS}} & \cdots & x_N^{\text{HMS}} & f(x^{\text{HMS}})
\end{bmatrix}
\]  

(2)

3.3. Improvise a new harmony

The new solution vector, \( x' = (x'_1, x'_2, \ldots, x'_{N-1}, x'_N) \), is improvised based on three operators: memory consideration, random consideration, and pitch adjustment. The memory consideration, which makes use of accumulative search from HM vectors (i.e. same functionality as the crossover operator in GA). The random consideration, which is used to diversify the new solution vector (i.e. same functionality as the mutation operator in GA). The pitch adjustment, which is responsible for the local improvement (i.e. same functionality as move in local search). The three operators work as follows:

3.3.1. Memory consideration

In this operator, the value of the decision variable \( x'_j \) for the new solution vector \( x' \) is selected from any of the historical values in the specified HM, \( x'_j \in R_j \), where \( R_j = \{x^i_j \mid i = 1, 2, \ldots, \text{HMS}\} \) with a probability of HMCR, where HMCR \((0, 1)\).

3.3.2. Random consideration

This operator randomly selects a value for the decision variable \( x'_j \) from its feasible range \( X_j \) with probability \( 1-\text{HMCR} \). The memory consideration and random consideration operators select the value of \( x'_j \) as follows:

\[
x'_j \leftarrow \begin{cases} 
  R_j \text{ with a probability } \text{HMCR}, \\
  X_j \text{ with a probability } (1-\text{HMCR}). 
\end{cases}
\]

3.3.3. Pitch adjustment

Every decision variable \( x'_j \) obtained by the memory consideration is adjusted to its neighboring value with probability PAR, where PAR \((0, 1)\), as follows:

\[
\text{pitch adjustment for } x'_j? \leftarrow \begin{cases} 
  \text{Yes} \text{ with a probability } \text{PAR}, \\
  \text{No} \text{ with a probability } (1-\text{PAR}). 
\end{cases}
\]
Harmony Search with Novel Selection Methods

If the pitch adjustment decision for $x_j'$ is Yes, the value of the decision variable $x_j'$ is modified to its neighboring value as follows:

$$x_j'(k) = v_{j,k \pm m},$$

where $x_j'$ is assigned with value $v_{j,k}$, that is, the $k$th element in $X_j$. $m$ is the neighboring index, $m \in \mathbb{Z}^+$. 

3.4. Update HM

If the new harmony $x'$ is better than the worst harmony $x_{\text{worst}}$ in the HM, the new harmony is included in the HM and the existing worst harmony is excluded from the HM.

3.5. Check the stopping criterion

Based on NI (maximum NI), Steps 3.3–3.5 of HSA are repeated.

4. HSA for NRP

Experiments with the basic HSA for NRP using INRC2010 dataset show that the algorithm suffers from the slow of convergence especially for the hardest instances. This is because the basic HSA not considered the natural rule of Darwins survival-of-the-fittest. In this paper, we used a set of selection methods inspired from other evolutionary techniques to replace the original random selection method in memory consideration of HSA, in the hope to speed up convergence. HSA for NRP has five main steps will be described below as follows.

4.1. Initialize the parameters of NRP and HSA

The solution vector (or roster in NRP) is represented as a vector of allocations (or decision variables), i.e. $x = (x_1, x_2, \ldots, x_{N-1}, x_N)$, where each allocation $x_j$ assigned four values (Nurse number, Day number, Shift number, and MCF) as shown in Table 3. The Memory Consideration Flag (MCF) is a binary variable that sets to one when the current allocation is assigned by the memory consideration or zero otherwise. This roster should be evaluated using the objective function (see (1)).

<table>
<thead>
<tr>
<th>Allocation</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>...</th>
<th>$x_{N-1}$</th>
<th>$x_N$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nurse</td>
<td>1</td>
<td>4</td>
<td>13</td>
<td>8</td>
<td>...</td>
<td>13</td>
<td>1</td>
</tr>
<tr>
<td>Day</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>7</td>
<td>...</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>Shift</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>...</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>MCF</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 3. Roster $x$ representation.
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 (all these parameters as discussed in Sec. 2). Furthermore, the parameters of HSA are also initialized in this step which includes HMS, HMCR, PAR, and NI.

4.2. Initialize HM

In this paper, we used a heuristic ordering (Burke et al., 2008) to construct the initial feasible rosters and stored them in ascending order in HM based on objective function value (see example (3)). The heuristic ordering method allocates different nurses in a roster as follows: this method sorts the different shifts in ascending order based on their difficulty level, noting that the lowest weekly nurses demand is the highest difficulty (see Table 4). Then the required nurses of the ordered shifts will be assigned starting from the most difficult and ending with the easiest. Furthermore, the worst roster \( x^{\text{worst}} \) (i.e. the roster with the highest penalty value) in HM is determined.

\[
\text{HM} = \begin{bmatrix}
(30, 0, 2, 1) & (1, 0, 0, 1) & \cdots & (30, 14, 1, 0) \\
(1, 0, 1, 1) & (2, 0, 2, 0) & \cdots & (5, 14, 1, 0) \\
(0, 0, 0, 0) & (30, 0, 1, 1) & \cdots & (13, 3, 2, 1)
\end{bmatrix}.
\]

4.3. Improvise a new harmony

The new roster, \( x' = (x'_1, x'_2, \ldots, x'_{N-1}, x'_N) \), is improvised based on the three operators: \textit{memory consideration}, \textit{random consideration}, and \textit{pitch adjustment}. It must be feasible (i.e. it must be satisfied all hard constraints). In contrast, if the improvisation process fails to improvise a feasible roster the repair procedure will be triggered. The three rules of the improvisation process work as follows.

<table>
<thead>
<tr>
<th>Shift type</th>
<th>Weekly nurses demand</th>
<th>Sum of demand</th>
<th>Ordering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day Shift (D)</td>
<td>Mon 7   Tue 7   Wed 6   Thu 7   Fri 5   Sat 5   Sun 4</td>
<td>44</td>
<td>4</td>
</tr>
<tr>
<td>Early Shift (E)</td>
<td>Mon 3   Tue 3   Wed 3   Thu 3   Fri 3   Sat 3   Sun 21</td>
<td>30</td>
<td>3</td>
</tr>
<tr>
<td>Late Shift (L)</td>
<td>Mon 5   Tue 5   Wed 4   Thu 5   Fri 3   Sat 3   Sun 30</td>
<td>30</td>
<td>3</td>
</tr>
<tr>
<td>Night Shift (N)</td>
<td>Mon 2   Tue 2   Wed 2   Thu 2   Fri 1   Sat 1   Sun 12</td>
<td>12</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 4. Ordering of shifts based on heuristic ordering method.
4.3.1. Memory consideration

The main concern of this study is to use a set of selection methods to replace the original random selection of this operator. In this operator, the allocation \( x'_{j} \) of the new roster \( x' \) will be assigned by a feasible value from a corresponding allocations \( x'_{j} \in R_{j} \), where \( R_{j} \subseteq \{ x_{j}^{i} | i = 1, 2, \ldots, \text{HMS} \} \) with probability of HMCR, where HMCR \( \in (0, 1) \). Basically, the elements of \( R_{j} \) are extracted from the rosters stored in HM by using Algorithm 2. However, if the \( R_{j} = \phi \), this means there is no feasible value for \( x'_{j} \) to make the new roster feasible. In such case, the Random Consideration is triggered to assign the value for \( x'_{j} \).

For example, the HM in (3) includes different allocations of three rosters \( x^{0} - x^{2} \). The first allocation \( x'_{1} \) is assigned value from \( R_{1} = \{(30, 0, 2, 1), (1, 0, 1, 1), (0, 0, 0, 0)\} \). Practically, all elements in \( R_{1} \) are feasible for \( x'_{1} \), due to this the first allocation takes a value, assuming that \( x'_{1} \) is assigned with the first value \((30, 0, 2, 1)\). It works the same manner in the second allocation \( x'_{2} \in R_{2} \), where \( R_{2} = \{(1, 0, 0, 1), (2, 0, 2, 0)\} \). Clearly, the value \((30, 0, 1, 1)\) is excluded from \( R_{2} \), and so there is a violation to the second hard constraints \( H_{2} \) (i.e. the nurse

\begin{verbatim}
Algorithm 2 Procedure of filling \( R_{j} \)
1: \( R_{j} = \phi \)
2: for \((i = 1 \text{ to HMS})\) do
3: \( \text{flag} = \text{false} \)
4: \( \text{count} = 0 \)
5: for \((k = 1 \text{ to } j-1)\) do
6: \( \text{if Nurse}(x_{i}^{j}) = \text{Nurse}(x_{k}^{j}) \land \text{Day}(x_{i}^{j}) = \text{Day}(x_{k}^{j}) \) then
7: \( \text{flag} = \text{true} \)
8: \( \text{GOTO} \ 2 \)
9: end if
10: end for
11: if \( \text{flag} = \text{false} \) then
12: for \((k = 1 \text{ to } j-1)\) do
13: \( \text{if} \ (\text{Day}(x_{i}^{j}) = \text{Day}(x_{k}^{j}) \land \text{Shift}(x_{i}^{j}) = \text{Shift}(x_{k}^{j})) \) then
14: \( \text{count} = \text{count} + 1 \)
15: end if
16: end for
17: \( s = \text{Shift}(x_{i}^{j}) \)
18: \( d = \text{Day}(x_{i}^{j}) \)
19: if \( \text{count} < \text{demand}_{d,a} \) then
20: \( R_{j} = R_{j} + x_{i}^{j} \)
21: end if
22: end if
23: end for
\end{verbatim}
is assigned two shifts \( s_2 \) and \( s_1 \) on the same day \( d_0 \), where the nurse can be assigned a maximum of one shift on the same day). The same process is used to assign values for other allocations. The different selection methods can be described as follows:

- **Random memory consideration selection (RMCS).** This is the original selection method of memory consideration operator that was introduced by Geem et al. (2001). Each roster in HM has the same chance (or probability) to give its value to a specific allocation \( x'_j \) in the new roster \( x' \) if the feasibility is achieved. In other words, at each iteration, the allocation \( x'_j \) will be assigned with a value \( x'_j \), where \( x'_j \) is selected randomly from \( R_j \subseteq \{ x_i \mid i = 1, 2, \ldots, \text{HMS} \} \).

- **Global-best memory consideration selection (GMCS).** This selection method was introduced by Kennedy and Eberhart (1995) derived from PSO. At each iteration, the best roster (\( x^{\text{best}} = x^1 \)) inside the HM has the total probability to give its value \( x^j \) for the current allocation \( x'_j \) in the new roster \( x' \), and the other roster probabilities in HM are zeros. Practically, if the feasibility of the \( x' \) is not achieved, the second best roster \( x^2 \) is used to give its value \( x^2_j \) for the allocation \( x'_j \), and so on until the last roster \( x^{\text{HMS}} \) is reached.

- **Proportional memory consideration selection (PMCS).** This selection method was introduced by Holland (1975). At each iteration, the fittest rosters have the higher probability to give value to a specific allocation \( x'_j \) in the new roster \( x' \), where the worst roster has the smaller probability. The probability of each roster is defined as follows:

\[
p(x^i) = \frac{\text{fit}(x^i)}{\sum_{s=1}^{\text{HMS}} \text{fit}(x^s)} \quad \forall i \in (1, 2, \ldots, \text{HMS}),
\]

where,

\[
\text{fit}(x^i) = \frac{1}{f(x^i)} \quad \text{where } i \in (1, 2, \ldots, \text{HMS}).
\]

The allocation \( x'_j \) will be assigned with a value of roster \( x^i \) using Algorithm 3, where the roster \( x^j \) is selected as follows: (i) the rosters in HM are mapped

---

**Algorithm 3** Proportional memory consideration selection

1. Set \( \text{rand} \sim U(0,1) \)
2. Set \( \text{Sum}_i \text{Prob}_i = 0 \)
3. for \( (i = 1 \text{ to } \text{HMS}) \) do
4. \( \text{Sum}_i \text{Prob}_i = \text{Sum}_i \text{Prob}_i + p(x^i) \)
5. if \( \text{Sum}_i \text{Prob}_i \geq \text{rand} \) then
6. \( x'_j = x^i_j \)
7. GOTO 9
8. end if
9. end for
to contiguous segments of a line, such that each roster’s segment represents its probability. Note that the summation of all probability values for all rosters in HM is unity, (ii) Generate a random value rand between 0 and 1, and (iii) select the roster whose segment spans the value of rand to give its value for the allocation $x'_j$ if feasible. If the feasibility is not achieved, repeat the steps (ii) and (iii).

- **Tournament memory consideration selection (TMCS).** Goldberg *et al.* (1989) proposed this selection method which operates as follows: (i) choose the size of the tournament $t$-size, where $1 \leq t$-size $\leq$ HMS, (ii) select the rosters randomly from the HM (in this study without duplications) to fill the tournament $t$, and (iii) select the best roster from the tournament $t$ to give its value to the allocation $x'_j$ in the improvisation process. The allocation $x'_j$ is assigned with the value of allocation $x^t_j$, where $x^t$ is the best roster in $t$ if the feasibility is achieved. If the feasibility is not achieved, then the selection process is repeated from scratch for that allocation $x'_j$. In GA, the popular size for $t$ fixed at each iteration is two, or selected randomly between one and HMS at each iteration.

- **Linear ranking memory consideration selection (LMCS).** This selection method was introduced by Baker (1985). At each iteration, each roster in HM is ranked, where the rank of best roster $x^1$ in HM is HMS (i.e. rank($x^1$) = HMS). In contrast, the rank of the worst roster $x^\text{HMS}$ in HM is one (i.e. rank($x^\text{HMS}$) = 1). The selection probability of each roster in HM is computed as follows:

$$p(x^i) = \frac{1}{\text{HMS}} \left( \min + \frac{(\max - \min)(\text{rank}(x^i) - 1)}{\text{HMS} - 1} \right),$$

where $\max + \min = 2$ and $1 \leq \max \leq 2$. Notably, the mechanism work of this selection method is like the proportional selection. Furthermore, the main difference between this selection and the proportional selection is that the proportional selection depends on the fitness of the rosters in HM, whereas the linear rank selection depends on the rank of the rosters in HM. The roster with the best rank is used to give its value for the allocation $x'_j$, if feasible and its probability exceeds the predefined random selection probability value.

### 4.3.2. Random consideration

This operator randomly selects a value for allocation $x'_j$ from its feasible range $X_j$ with probability $1$-HMCR where the rules of heuristic ordering are considered. In practice, the elements of $X_j$ are concluded from the search space of NRP using Algorithm 4. It should be noted that $m$ is the total number of nurses allocated to maximum number of shifts $r$, during the scheduling period $b$.

### 4.3.3. Pitch adjustment

This operator adjusts the allocation $x'_j$ obtained 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.
M. A. Awadallah et al.

Algorithm 4 Procedure of filling $X_j$

```
1: for (d = 0 to b-1) do
2:   for (s=0 to r-1) do
3:     for (n=0 to m-1) do
4:       if skill(n) ≥ skill(s) then
5:         for (k = 1 to j-1) do
6:           if (n = Nurse($x'_k$) ∧ d = Day($x'_k$)) then
7:             GOTO 3
8:         else
9:           $X_j = X_j + (n,d,s,0)$
10:       end if
11:     end for
12:   end if
13: end for
14: end for
15: end for
```

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 its violations such as ($S_1$−$S_3$).

This operator adjusts the allocation $x'_j$ selected by the memory consideration
when the MCF($x'_j$) = “one” to its neighboring value with probability PAR where
PAR ∈ (0, 1) as follows:

pitch adjustment for $x'_j$? ← \[
\begin{cases}
  \text{Yes} & \text{with a probability } \frac{PAR}{8}, \\
  \text{No} & \text{with a probability } (1 - \frac{PAR}{8}) 
\end{cases}
\]

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

```
x'_j ← \begin{align*}
  \text{MoveOneShift} & : 0 ≤ \text{rnd} ≤ \frac{PAR}{8}, \\
  \text{SwapOneShift} & : \frac{PAR}{8} < \text{rnd} ≤ 2 \times \frac{PAR}{8}, \\
  \text{TokenRingMove} & : 2 \times \frac{PAR}{8} < \text{rnd} ≤ 3 \times \frac{PAR}{8}, \\
  \text{Swap2Shifts} & : 3 \times \frac{PAR}{8} < \text{rnd} ≤ 4 \times \frac{PAR}{8}, \\
  \text{CrossMove} & : 4 \times \frac{PAR}{8} < \text{rnd} ≤ 5 \times \frac{PAR}{8}, \\
  \text{MoveWeekend} & : 5 \times \frac{PAR}{8} < \text{rnd} ≤ 6 \times \frac{PAR}{8}, \\
  \text{SwapConsecutive2Days} & : 6 \times \frac{PAR}{8} < \text{rnd} ≤ 7 \times \frac{PAR}{8}, \\
  \text{SwapConsecutive3Days} & : 7 \times \frac{PAR}{8} < \text{rnd} ≤ \frac{PAR}{8}, \\
  \text{Do Nothing} & : PAR < \text{rnd} ≤ 1,
\end{align*}
```
where rnd is generated randomly between 0 and 1. The eight pitch adjustment neighborhood structures are designed to run as follows:

(1) **MoveOneShift pitch adjustment.** This neighborhood is used to transfer the shift of specific allocation from the current nurse to another nurse selected randomly with probability \([0, \text{PAR}/8]\).

(2) **SwapOneShift pitch adjustment.** This neighborhood is used to exchange shifts of two allocations, while the both allocations in the same day. This procedure is triggered with probability \((\text{PAR}/8, \text{PAR}/4]\).

(3) **TokenRingMove pitch adjustment.** This neighborhood includes two phases to solve the violations of two soft constraints \(S_4\) and \(S_5\). Firstly, transfer the shift of specific allocation from the current nurse to another nurse selected randomly if the soft constraint \(S_4\) is not achieved. Secondly, exchange shift of the current allocation with another allocation selected randomly to solve the violations of the soft constraint \(S_5\) if not achieved. This procedure is triggered with probability \((\text{PAR}/4, 3\times\text{PAR}/8]\).

(4) **Swap2Shifts pitch adjustment.** This neighborhood is used to exchange the shifts of four allocations, while each pair of allocations in the same day. This procedure is triggered with probability \((3 \times \text{PAR}/8, \text{PAR}/2]\).

(5) **CrossMove pitch adjustment.** This neighborhood is used to make double move shifts between two nurses. Firstly, transfer the shift of specific allocation from one nurse to another nurse. Secondly, transfer another shift from the second nurse to first nurse. This procedure is triggered with probability \((\text{PAR}/2, 5 \times \text{PAR}/8]\).

(6) **MoveWeekend pitch adjustment.** This neighborhood is used to transfer the shifts of the weekend from one nurse to another free nurse selected randomly with probability \((5 \times \text{PAR}/8, 6 \times \text{PAR}/8]\).

(7) **SwapConsecutive2Days pitch adjustment.** This neighborhood is used to exchange pattern of shifts of two days between two nurses with probability \((6 \times \text{PAR}/8, 7 \times \text{PAR}/8]\).

(8) **SwapConsecutive3Days pitch adjustment.** This neighborhood is used to exchange pattern of shifts of three days between two nurses with probability \((7 \times \text{PAR}/8, \text{PAR}\].

The pseudo code of the new improvisation process is provided in Algorithm 5. In this paper, any pitch adjustment procedures 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 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 as follows: Firstly, identify all allocations that are not scheduled in the new roster; secondly, identify the day(s), where the nurses demand are not completely
Algorithm 5 New improvisation process

1: $x' = \phi$ \hspace{1cm} \{new roster\}
2: for $j = 1$ to $N$ do
3: \hspace{1cm} $R_j$ is defined using algorithm 2
4: \hspace{1cm} MCF($x'_j$) = 0
5: \hspace{1cm} if ($U(0,1) \leq$ HMCR) then
6: \hspace{1cm} \hspace{1cm} if $R_j = \phi$ then
7: \hspace{1cm} \hspace{1cm} \hspace{1cm} $x'_j \in X_j$ \hspace{1cm} \{random consideration\}
8: \hspace{1cm} \hspace{1cm} else
9: \hspace{1cm} \hspace{1cm} \hspace{1cm} $x'_j \in R_j$ \hspace{1cm} \{memory consideration\}
10: \hspace{1cm} \hspace{1cm} end if
11: \hspace{1cm} end if
12: else
13: \hspace{1cm} $x'_j \in X_j$ \hspace{1cm} \{random consideration\}
14: end if
15: end for
16: for $j = 1$ to $N$ do
17: \hspace{1cm} if MCF($x'_j$) = 1 then
18: \hspace{1cm} \hspace{1cm} if ($U(0,1) \leq$ PAR) then
19: \hspace{1cm} \hspace{1cm} \hspace{1cm} pitch adjustment for ($x'_j$) \hspace{1cm} \{pitch adjustment\}
20: \hspace{1cm} \hspace{1cm} end if
21: \hspace{1cm} end if
22: end for

scheduled in the new roster; and thirdly, for each day identified, copy the allocations of the same day from the previous or next weeks.

4.4. Update HM

After a new roster $x'$ is improvised, the HM will be updated by the natural rule of Darwins principle “survival-of-the-fittest” between the new roster $x'$ and the worst roster $x_{\text{worst}}$ in HM. That is, the new roster $x'$ replaces the worst roster $x_{\text{worst}}$ in HM. Furthermore, reordering the rosters in HM in ascending order will be considered.

4.5. Check the stopping criterion

If the stopping criterion (NI) is satisfied, computation is terminated. Otherwise, Steps 4.3–4.5 are repeated.

5. Experimental Results

In this section, the performance of HSA with different memory consideration selection methods for NRP is evaluated using standard datasets published by the
first International Nurse Rostering Competition INRC2010. The proposed HSA 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 parameters settings are the same for different selection methods due to study the effectiveness of different selection methods under the same environment, which includes HMS = 10, HMCR = 0.99, PAR = 0.7, and NI = 100,000. It is worthy of note that, the choices of these values are based on the best parameter settings achieved in Awadallah et al. (2013).

5.1. INRC2010 dataset

The dataset established in INRC2010 is classified into three tracks: sprint, medium, and long, which are varied in 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 is the easiest, which consists of 33 instances are divided into 10 early, 10 hidden, 10 late, and 3 hint. The medium track is more complex than sprint track. It includes 18 instances, which categorized to 5 early, 5 hidden, 5 late, and 3 hint. The long track is the hardest, which includes 18 instances divided like medium track instances. Tables 5-7 showed the characteristics of sprint, medium, long tracks instances respectively. The features of each instance included in these tables are: the name; number of shifts; number of nurses available; number of different contracts; number of different skills for all nurses; number of unwanted patterns; number of weekend days such as two days (i.e., Saturday–Sunday) or three days (i.e., Friday–Saturday–Sunday); the existence of nurse preferences (i.e., Day-off, Shift-off); and the scheduling period. Notably, number of shifts in some of the instances is not fixed. In other words, number of shifts for each day during the scheduling period is not same. Example, sprint_hidden04 some days divided to three shifts and other days divided to four shifts. Furthermore, the weekend days are not fixed for all nurses.

Table 5. The characteristics of sprint track instances.

<table>
<thead>
<tr>
<th>Instance</th>
<th>Shifts</th>
<th>Nurses</th>
<th>Contracts</th>
<th>Skills</th>
<th>Patterns</th>
<th>Weekends</th>
<th>Day- off</th>
<th>Shift- off</th>
<th>Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>sprint_early</td>
<td>01–10</td>
<td>4</td>
<td>10</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>✓</td>
<td>✓±25/01/2010</td>
</tr>
<tr>
<td>sprint_hidden</td>
<td>01, 02</td>
<td>3</td>
<td>10</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>✓</td>
<td>✓±25/06/2010</td>
</tr>
<tr>
<td>sprint_hidden</td>
<td>03, 05, 08</td>
<td>4</td>
<td>10</td>
<td>3</td>
<td>1</td>
<td>8</td>
<td>2</td>
<td>✓</td>
<td>✓±25/06/2010</td>
</tr>
<tr>
<td>sprint_hidden</td>
<td>04, 09</td>
<td>3, 4</td>
<td>10</td>
<td>3</td>
<td>1</td>
<td>8</td>
<td>2</td>
<td>✓</td>
<td>✓±25/06/2010</td>
</tr>
<tr>
<td>sprint_hidden</td>
<td>06-07</td>
<td>3</td>
<td>10</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>✓</td>
<td>✓±25/01/2010</td>
</tr>
<tr>
<td>sprint_hidden</td>
<td>10</td>
<td>4</td>
<td>10</td>
<td>3</td>
<td>1</td>
<td>8</td>
<td>2</td>
<td>✓</td>
<td>✓±25/01/2010</td>
</tr>
<tr>
<td>sprint_early</td>
<td>01, 03-05</td>
<td>4</td>
<td>10</td>
<td>3</td>
<td>1</td>
<td>8</td>
<td>2</td>
<td>✓</td>
<td>✓±25/01/2010</td>
</tr>
<tr>
<td>sprint_early</td>
<td>02</td>
<td>3</td>
<td>10</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>✓</td>
<td>✓±25/01/2010</td>
</tr>
<tr>
<td>sprint_late</td>
<td>06, 07, 10</td>
<td>4</td>
<td>10</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>✓</td>
<td>✓±25/01/2010</td>
</tr>
<tr>
<td>sprint_late</td>
<td>08</td>
<td>4</td>
<td>10</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>×</td>
<td>×±25/01/2010</td>
</tr>
<tr>
<td>sprint_late</td>
<td>09</td>
<td>4</td>
<td>10</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>2, 3</td>
<td>×</td>
<td>×±25/01/2010</td>
</tr>
<tr>
<td>sprint_hidden</td>
<td>01, 03</td>
<td>4</td>
<td>10</td>
<td>3</td>
<td>1</td>
<td>8</td>
<td>2</td>
<td>✓</td>
<td>✓±25/01/2010</td>
</tr>
<tr>
<td>sprint_hidden</td>
<td>02</td>
<td>4</td>
<td>10</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>✓</td>
<td>✓±25/01/2010</td>
</tr>
</tbody>
</table>
Table 6. The characteristics of medium track instances.

<table>
<thead>
<tr>
<th>Instance Shifts</th>
<th>Nurses</th>
<th>Contracts</th>
<th>Skills</th>
<th>Patterns</th>
<th>Weekends</th>
<th>Day-Shift-Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>medium early 01-05</td>
<td>4</td>
<td>31</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>✓ ✓ 1-28/01/2010</td>
</tr>
<tr>
<td>medium hidden 01-04</td>
<td>5</td>
<td>30</td>
<td>4</td>
<td>2</td>
<td>9</td>
<td>✓ ✓ 1-28/06/2010</td>
</tr>
<tr>
<td>medium late 05</td>
<td>5</td>
<td>30</td>
<td>4</td>
<td>1</td>
<td>9</td>
<td>✓ ✓ 1-28/06/2010</td>
</tr>
<tr>
<td>medium late 01</td>
<td>4</td>
<td>30</td>
<td>4</td>
<td>1</td>
<td>7</td>
<td>✓ ✓ 1-28/01/2010</td>
</tr>
<tr>
<td>medium late 02-04</td>
<td>4</td>
<td>30</td>
<td>3</td>
<td>1</td>
<td>7</td>
<td>✓ ✓ 1-28/01/2010</td>
</tr>
<tr>
<td>medium late 03</td>
<td>4</td>
<td>30</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>✓ ✓ 1-28/01/2010</td>
</tr>
<tr>
<td>medium late 05</td>
<td>5</td>
<td>30</td>
<td>4</td>
<td>2</td>
<td>7</td>
<td>✓ ✓ 1-28/01/2010</td>
</tr>
<tr>
<td>medium hint 01-03</td>
<td>4</td>
<td>30</td>
<td>4</td>
<td>1</td>
<td>7</td>
<td>✓ ✓ 1-28/01/2010</td>
</tr>
<tr>
<td>medium hint 02</td>
<td>4</td>
<td>30</td>
<td>3</td>
<td>1</td>
<td>7</td>
<td>✓ ✓ 1-28/01/2010</td>
</tr>
</tbody>
</table>

Table 7. The characteristics of long track instances.

<table>
<thead>
<tr>
<th>Instance Shifts</th>
<th>Nurses</th>
<th>Contracts</th>
<th>Skills</th>
<th>Patterns</th>
<th>Weekends</th>
<th>Day-Shift-Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>long early 01-05</td>
<td>5</td>
<td>49</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>✓ ✓ 1-28/01/2010</td>
</tr>
<tr>
<td>long hidden 01-04</td>
<td>5</td>
<td>50</td>
<td>3</td>
<td>2</td>
<td>9</td>
<td>✓ ✓ 1-28/06/2010</td>
</tr>
<tr>
<td>long hidden 05</td>
<td>5</td>
<td>50</td>
<td>4</td>
<td>2</td>
<td>9</td>
<td>✓ ✓ 1-28/06/2010</td>
</tr>
<tr>
<td>long late 01-03</td>
<td>5</td>
<td>50</td>
<td>3</td>
<td>2</td>
<td>9</td>
<td>✓ ✓ 1-28/01/2010</td>
</tr>
<tr>
<td>long late 02-04</td>
<td>5</td>
<td>50</td>
<td>4</td>
<td>2</td>
<td>9</td>
<td>✓ ✓ 1-28/01/2010</td>
</tr>
<tr>
<td>long hint 01</td>
<td>5</td>
<td>50</td>
<td>3</td>
<td>2</td>
<td>9</td>
<td>✓ ✓ 1-28/01/2010</td>
</tr>
<tr>
<td>long hint 02-03</td>
<td>5</td>
<td>50</td>
<td>3</td>
<td>2</td>
<td>7</td>
<td>✓ ✓ 1-28/01/2010</td>
</tr>
</tbody>
</table>

in the same instance. For example, the weekend days for some nurse are Saturday–Sunday, while the weekend days for the other nurses are Friday–Saturday–Sunday in the same instance.

5.2. The effect of tournament size $t$ on behavior of HSA with tournament memory consideration

Table 8 provides the results of the HSA with tournament memory consideration on INRC2010 dataset using different tournament size (i.e. $t = 2$, $t = 5$, $t = 10$ = HMS, $t \sim U(1, \text{HMS})$). The numbers in Table 8 is referred to the value of the objective function which was formalized in (1), where the lowest value is the best. For each instance, the best (B.), mean (M.), worst (W.), and standard deviation (Std.) of the 10 runs are recorded. The best result is highlighted in bold. Experimentally, the behavior of HSA is improving as the tournament size $t$ increases. This is because the tournament with larger size $t$ has higher chance to include the best solution in the HM. Apparently, the HSA with $t = 10 = \text{HMS}$ achieved the best results in 42 out 69 instances of INRC2010 dataset. This is due to the tournament at all times includes the best solution in HM, and uses this solution in the improvisation process. In other words, the selection mechanism of this method is similar to Global-best selection of PSO, where the new solution inherits the characteristics of the best
Table 8. The results of HSA using various tournament size $t$.

<table>
<thead>
<tr>
<th>Instance</th>
<th>$t = 2$</th>
<th>$t = 5$</th>
<th>$t = 10$</th>
<th>$t \sim U(1, \text{HMS})$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>B.</td>
<td>M.</td>
<td>W.</td>
<td>S.</td>
</tr>
<tr>
<td>sprint_early01</td>
<td>58</td>
<td>60.8</td>
<td>75.0</td>
<td>5.0</td>
</tr>
<tr>
<td>sprint_early02</td>
<td>58</td>
<td>61.7</td>
<td>64.1</td>
<td>1.7</td>
</tr>
<tr>
<td>sprint_early03</td>
<td>54</td>
<td>63.8</td>
<td>88.9</td>
<td>1.3</td>
</tr>
<tr>
<td>sprint_early04</td>
<td>64</td>
<td>80.5</td>
<td>87.9</td>
<td>14.0</td>
</tr>
<tr>
<td>sprint_early05</td>
<td>59</td>
<td>61.2</td>
<td>63.1</td>
<td>1.3</td>
</tr>
<tr>
<td>sprint_early06</td>
<td>56</td>
<td>62.6</td>
<td>75.4</td>
<td>6.4</td>
</tr>
<tr>
<td>sprint_early07</td>
<td>59</td>
<td>68.1</td>
<td>82.9</td>
<td>9.2</td>
</tr>
<tr>
<td>sprint_early08</td>
<td>58</td>
<td>67.7</td>
<td>85.8</td>
<td>10.7</td>
</tr>
<tr>
<td>sprint_early09</td>
<td>57</td>
<td>64.1</td>
<td>77.6</td>
<td>6.8</td>
</tr>
<tr>
<td>sprint_early10</td>
<td>56</td>
<td>63.3</td>
<td>80.5</td>
<td>9.6</td>
</tr>
<tr>
<td>sprint_hidden01</td>
<td>82</td>
<td>113.8</td>
<td>157.8</td>
<td>26.0</td>
</tr>
<tr>
<td>sprint_hidden02</td>
<td>37</td>
<td>51.2</td>
<td>79.4</td>
<td>14.8</td>
</tr>
<tr>
<td>sprint_hidden03</td>
<td>77</td>
<td>118.4</td>
<td>144.1</td>
<td>25.1</td>
</tr>
<tr>
<td>sprint_hidden04</td>
<td>146</td>
<td>163.1</td>
<td>179.9</td>
<td>9.7</td>
</tr>
<tr>
<td>sprint_hidden05</td>
<td>67</td>
<td>95.3</td>
<td>148.7</td>
<td>27.3</td>
</tr>
<tr>
<td>sprint_hidden06</td>
<td>627</td>
<td>793</td>
<td>940.8</td>
<td>89.4</td>
</tr>
<tr>
<td>sprint_hidden07</td>
<td>235</td>
<td>486.7</td>
<td>649.1</td>
<td>107.4</td>
</tr>
<tr>
<td>sprint_hidden08</td>
<td>543</td>
<td>664.7</td>
<td>815.8</td>
<td>92.1</td>
</tr>
<tr>
<td>sprint_hidden09</td>
<td>864</td>
<td>937.9</td>
<td>1009.9</td>
<td>45.4</td>
</tr>
<tr>
<td>sprint_hidden10</td>
<td>567</td>
<td>724.8</td>
<td>972.1</td>
<td>121.1</td>
</tr>
<tr>
<td>sprint_late01</td>
<td>52</td>
<td>85.7</td>
<td>120.4</td>
<td>21.4</td>
</tr>
<tr>
<td>sprint_late02</td>
<td>52</td>
<td>86.9</td>
<td>115.2</td>
<td>20.8</td>
</tr>
<tr>
<td>sprint_late03</td>
<td>60</td>
<td>93.8</td>
<td>113.1</td>
<td>19.4</td>
</tr>
<tr>
<td>sprint_late04</td>
<td>229</td>
<td>338</td>
<td>444.4</td>
<td>66.6</td>
</tr>
<tr>
<td>sprint_late05</td>
<td>51</td>
<td>83.6</td>
<td>105.2</td>
<td>19.2</td>
</tr>
<tr>
<td>sprint_late06</td>
<td>48</td>
<td>60.4</td>
<td>86.1</td>
<td>13.5</td>
</tr>
<tr>
<td>sprint_late07</td>
<td>62</td>
<td>137.6</td>
<td>210.4</td>
<td>47.0</td>
</tr>
<tr>
<td>sprint_late08</td>
<td>22</td>
<td>32.2</td>
<td>55.1</td>
<td>10.4</td>
</tr>
<tr>
<td>sprint_late09</td>
<td>17</td>
<td>43.1</td>
<td>109.2</td>
<td>28.7</td>
</tr>
<tr>
<td>sprint_late10</td>
<td>56</td>
<td>123.8</td>
<td>294.4</td>
<td>57.3</td>
</tr>
</tbody>
</table>

(Continued)
<table>
<thead>
<tr>
<th>Instance</th>
<th>t = 2</th>
<th>t = 5</th>
<th>t = 10</th>
<th>t ~ U(1, HMS)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>B. M. W. S.</td>
<td>B. M. W. S.</td>
<td>B. M. W. S.</td>
<td>B. M. W. S.</td>
</tr>
<tr>
<td>spring_hint01</td>
<td>240 331.2 407 60.1</td>
<td>91 107.6 137 16.4</td>
<td>99 110.6 119 6.9</td>
<td>96 123.8 288 58.0</td>
</tr>
<tr>
<td>spring_hint02</td>
<td>60 102.3 142 26.2</td>
<td>57 68.4 74 5.6</td>
<td>60 70.2 77 5.6</td>
<td>60 69.5 86 6.8</td>
</tr>
<tr>
<td>spring_hint03</td>
<td>219 287.9 407 59.3</td>
<td>78 92.3 120 14.8</td>
<td>82 92.6 106 8.2</td>
<td>75 89.4 98 7.8</td>
</tr>
<tr>
<td>medium_early01</td>
<td>448 455.4 469 6.8</td>
<td>360 396.3 425 18.5</td>
<td>263 320.3 374 43.3</td>
<td>355 377.2 399 14.6</td>
</tr>
<tr>
<td>medium_early02</td>
<td>440 453.1 467 8.8</td>
<td>359 385.7 425 20.0</td>
<td>279 320.4 398 38.0</td>
<td>362 388.9 418 21.2</td>
</tr>
<tr>
<td>medium_early03</td>
<td>439 416.8 457 6.5</td>
<td>328 373.5 407 26.2</td>
<td>259 323.2 371 37.6</td>
<td>354 378.5 405 17.1</td>
</tr>
<tr>
<td>medium_early04</td>
<td>444 457 474 9.0</td>
<td>356 393 431 19.5</td>
<td>257 306.2 381 39.8</td>
<td>330 379.4 416 26.3</td>
</tr>
<tr>
<td>medium_early05</td>
<td>509 522 531 6.6</td>
<td>421 449.5 472 19.2</td>
<td>327 367 414 29.4</td>
<td>428 456.5 490 22.8</td>
</tr>
<tr>
<td>medium_medium01</td>
<td>2588 2652.5 2745 60.1</td>
<td>1272 1685.6 1957 219.4</td>
<td>235 419.9 1601 416.5</td>
<td>1736 1891.3 2077 123.3</td>
</tr>
<tr>
<td>medium_medium02</td>
<td>2039 2115.9 2184 53.1</td>
<td>1163 1305.2 1560 125.7</td>
<td>345 439.3 851 148.5</td>
<td>1149 1355.6 1526 120.5</td>
</tr>
<tr>
<td>medium_medium03</td>
<td>607 644.5 663 18.2</td>
<td>426 487 541 36.6</td>
<td>80 109.9 223 40.7</td>
<td>437 472.4 534 30.1</td>
</tr>
<tr>
<td>medium_medium04</td>
<td>608 634.4 656 13.0</td>
<td>440 480.4 560 41.5</td>
<td>127 192.1 372 93.4</td>
<td>428 460.2 512 28.7</td>
</tr>
<tr>
<td>medium_medium05</td>
<td>2349 2438.8 2494 45.6</td>
<td>1281 1337.7 1825 189.1</td>
<td>269 303.8 359 29.2</td>
<td>1360 1644.9 2098 202.7</td>
</tr>
<tr>
<td>medium_medium01</td>
<td>1629 1813.7 1906 77.8</td>
<td>806 1114.3 1360 159.1</td>
<td>268 432.3 1093 274.8</td>
<td>858 1024.1 1237 122.0</td>
</tr>
<tr>
<td>medium_medium02</td>
<td>402 426.7 450 13.4</td>
<td>150 224.1 288 36.7</td>
<td>74 123.4 236 66.8</td>
<td>197 265 306 39.6</td>
</tr>
<tr>
<td>medium_medium03</td>
<td>512 533.5 551 12.6</td>
<td>174 295.9 365 55.6</td>
<td>68 96.6 237 50.0</td>
<td>212 333.8 393 57.6</td>
</tr>
<tr>
<td>medium_medium04</td>
<td>425 441.5 455 8.3</td>
<td>137 241.9 307 61.1</td>
<td>78 90.3 103 7.0</td>
<td>209 280.1 373 61.3</td>
</tr>
<tr>
<td>medium_medium05</td>
<td>1667 1830.5 1957 79.1</td>
<td>751 1086.9 1379 178.4</td>
<td>244 311.1 698 137.1</td>
<td>894 1074.3 1437 140.3</td>
</tr>
<tr>
<td>medium_medium01</td>
<td>591 617.8 661 19.2</td>
<td>359 411.4 457 36.3</td>
<td>83 112.8 282 59.9</td>
<td>359 420.7 479 40.5</td>
</tr>
<tr>
<td>medium_medium02</td>
<td>1922 2029.6 2117 57.8</td>
<td>931 1253.6 1685 200.5</td>
<td>191 209.1 234 16.5</td>
<td>1034 1208.4 1548 156.5</td>
</tr>
<tr>
<td>medium_medium03</td>
<td>4032 4924.3 5507 456.2</td>
<td>2734 3058.6 3501 249.7</td>
<td>219 315.9 872 196.7</td>
<td>2399 2912.6 3332 322.4</td>
</tr>
</tbody>
</table>
### Table 8. (Continued)

<table>
<thead>
<tr>
<th>Instance</th>
<th>$t = 2$</th>
<th></th>
<th>$t = 5$</th>
<th></th>
<th>$t = 10$</th>
<th></th>
<th>$t \sim U(1, HMS)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>B.</td>
<td>M.</td>
<td>W.</td>
<td>S.</td>
<td>B.</td>
<td>M.</td>
<td>W.</td>
</tr>
<tr>
<td>long.early01</td>
<td>493</td>
<td>509.2</td>
<td>530</td>
<td>13.0</td>
<td>325</td>
<td>377.5</td>
<td>410</td>
</tr>
<tr>
<td>long.early02</td>
<td>544</td>
<td>558.8</td>
<td>567</td>
<td>9.0</td>
<td>396</td>
<td>435.4</td>
<td>479</td>
</tr>
<tr>
<td>long.early03</td>
<td>476</td>
<td>497</td>
<td>525</td>
<td>14.6</td>
<td>324</td>
<td>374</td>
<td>407</td>
</tr>
<tr>
<td>long.early04</td>
<td>587</td>
<td>608.5</td>
<td>619</td>
<td>9.6</td>
<td>431</td>
<td>484.8</td>
<td>552</td>
</tr>
<tr>
<td>long.early05</td>
<td>559</td>
<td>582.4</td>
<td>598</td>
<td>15.4</td>
<td>385</td>
<td>447.7</td>
<td>518</td>
</tr>
<tr>
<td>long.hidden01</td>
<td>414</td>
<td>421.7</td>
<td>432.7</td>
<td>57.6</td>
<td>2611</td>
<td>2873.8</td>
<td>3321</td>
</tr>
<tr>
<td>long.hidden02</td>
<td>841</td>
<td>862.5</td>
<td>889</td>
<td>14.6</td>
<td>598</td>
<td>673.6</td>
<td>749</td>
</tr>
<tr>
<td>long.hidden03</td>
<td>779</td>
<td>801.9</td>
<td>828</td>
<td>16.4</td>
<td>385</td>
<td>504.2</td>
<td>613</td>
</tr>
<tr>
<td>long.hidden04</td>
<td>734</td>
<td>753.4</td>
<td>787</td>
<td>15.2</td>
<td>423</td>
<td>501.4</td>
<td>587</td>
</tr>
<tr>
<td>long.hidden05</td>
<td>914</td>
<td>936.9</td>
<td>965</td>
<td>17.8</td>
<td>509</td>
<td>587.5</td>
<td>724</td>
</tr>
<tr>
<td>long.hint01</td>
<td>3649</td>
<td>3874.5</td>
<td>4065</td>
<td>138.1</td>
<td>2161</td>
<td>2721.6</td>
<td>2958</td>
</tr>
<tr>
<td>long.hint02</td>
<td>3868</td>
<td>4010.7</td>
<td>4209</td>
<td>98.7</td>
<td>2376</td>
<td>2761.6</td>
<td>2903</td>
</tr>
<tr>
<td>long.hint03</td>
<td>3480</td>
<td>3698.2</td>
<td>3896</td>
<td>127.4</td>
<td>2234</td>
<td>2632.6</td>
<td>3037</td>
</tr>
<tr>
<td>long.hint04</td>
<td>3760</td>
<td>3897.7</td>
<td>3993</td>
<td>60.2</td>
<td>2642</td>
<td>2792.2</td>
<td>3230</td>
</tr>
<tr>
<td>long.hint05</td>
<td>3110</td>
<td>3191.8</td>
<td>3315</td>
<td>60.4</td>
<td>2018</td>
<td>2232.7</td>
<td>2560</td>
</tr>
<tr>
<td>long.hint06</td>
<td>770</td>
<td>796.8</td>
<td>829</td>
<td>16.3</td>
<td>531</td>
<td>572</td>
<td>631</td>
</tr>
<tr>
<td>long.hint07</td>
<td>559</td>
<td>579.3</td>
<td>601</td>
<td>14.3</td>
<td>350</td>
<td>400.7</td>
<td>489</td>
</tr>
<tr>
<td>long.hint08</td>
<td>2905</td>
<td>3086.8</td>
<td>3176</td>
<td>74.4</td>
<td>1904</td>
<td>2109.6</td>
<td>2341</td>
</tr>
</tbody>
</table>

Harmony Search with Novel Selection Methods
solution in HM. It should be noted that the results of HSA with $t = 10 = \text{HMS}$ is used for comparison in Sec. 5.4.

5.3. **The effect of parameter $\text{max}$ on the behavior of HSA with linear rank memory consideration**

The results of HSA with linear rank memory consideration using various $\text{max}$ values (i.e. $\text{max} = 1$, $\text{max} = 1.3$, $\text{max} = 1.6$, $\text{max} = 1.9$) are provided in Table 9. The numbers in Table 9 refer to the value of the objective function which was formulated in (1), while the lowest value is the best. For each instance, the best (B.), mean (M.), worst (W.), and standard deviation (Std.) of the 10 runs are recorded. This is to study the behavior of HSA with various $\text{max}$ using INRC2010 dataset, where the best is highlighted in **bold**. Clearly, the best results achieved by HSA when the value of the parameter $\text{max}$ is close to 2. This is because the higher the value of $\text{max}$, the higher the rate of exploitation, and which is useful for solving NRP, where the search space for NRP is seems rugged. Experimentally, the HSA with $\text{max} = 1.9$ obtained the best results in 51 out of 69 instances of INRC2010 dataset. Note that the best results achieved by this variant HSA are used for comparison in Sec. 5.4.

5.4. **Comparing results of the harmony search-based algorithms**

The experimental results of HSA with the different memory consideration selection methods (i.e. RMCS, GMCS, PMCS, TMCS, and LMCS) are summarized in Table 10. The numbers in Table 10 indicate to the value of the objective function, where the lowest value is the best. For each selection method of each instance, the best (B.), mean (M.), worst (W.), standard deviation (Std.) of the 10 runs are recorded. Furthermore, the best result among the five memory consideration selection methods for each instance of INRC2010 dataset is highlighted in **bold**.

As shown in Table 10, for the sprint track instances results, it is difficult to decide which selection method achieved the best results. Whereas the RMCS achieved the best results in 11 instances, the LMCS in 10 instances, the PMCS in 8 instances, the TMCS in 6 instances, and the Global-best memory consideration selection in 5 instances out of 69 instances of INRC2010 dataset.

Similarly, for medium track instances as recorded in Table 10, the TMCS achieved the best results in 14 out of 18 instances. This is due to the nature of the NRP, where the search space of the medium track instances seems rugged and need more efforts to achieve good results. In contrast, the search space of NRP in the smallest instance is very simple. This is because of the limited number of variables and the small size of these instances. Naturally, the new roster in the TMCS inherits the characteristics of the best roster in the tournament, where the rosters in the tournament selected randomly from HM.

Furthermore, Table 10 provides the results of the various memory consideration selection methods for long track instances. Experimentally, the Global-best memory consideration and the TMCSs achieved the best results in all instances of INRC2010.
Table 9. The results of HSA using various max values.

<table>
<thead>
<tr>
<th>Instance</th>
<th>B.</th>
<th>M.</th>
<th>W.</th>
<th>S.</th>
<th>max = 1.0</th>
<th>B.</th>
<th>M.</th>
<th>W.</th>
<th>S.</th>
<th>max = 1.3</th>
<th>B.</th>
<th>M.</th>
<th>W.</th>
<th>S.</th>
<th>max = 1.6</th>
<th>B.</th>
<th>M.</th>
<th>W.</th>
<th>S.</th>
<th>max = 1.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>sprint early 01</td>
<td>58</td>
<td>59.2</td>
<td>60</td>
<td>0.9</td>
<td></td>
<td>58</td>
<td>59.2</td>
<td>60</td>
<td>0.8</td>
<td></td>
<td>58</td>
<td>59.1</td>
<td>61</td>
<td>1.1</td>
<td></td>
<td>58</td>
<td>59.8</td>
<td>61</td>
<td>0.9</td>
<td></td>
</tr>
<tr>
<td>sprint early 02</td>
<td>60</td>
<td>61.8</td>
<td>64</td>
<td>1.5</td>
<td></td>
<td>59</td>
<td>61.3</td>
<td>63</td>
<td>1.6</td>
<td></td>
<td>54</td>
<td>56.1</td>
<td>58</td>
<td>1.7</td>
<td></td>
<td>54</td>
<td>55.5</td>
<td>57</td>
<td>1.1</td>
<td></td>
</tr>
<tr>
<td>sprint early 03</td>
<td>54</td>
<td>56.5</td>
<td>60</td>
<td>2.0</td>
<td></td>
<td>54</td>
<td>56.1</td>
<td>58</td>
<td>1.7</td>
<td></td>
<td>62</td>
<td>67.4</td>
<td>83</td>
<td>7.1</td>
<td></td>
<td>62</td>
<td>64.2</td>
<td>66</td>
<td>1.1</td>
<td></td>
</tr>
<tr>
<td>sprint early 04</td>
<td>63</td>
<td>64.6</td>
<td>66</td>
<td>1.1</td>
<td></td>
<td>63</td>
<td>65.6</td>
<td>74</td>
<td>3.8</td>
<td></td>
<td>60</td>
<td>62.6</td>
<td>69</td>
<td>2.9</td>
<td></td>
<td>60</td>
<td>64.2</td>
<td>82</td>
<td>8.0</td>
<td></td>
</tr>
<tr>
<td>sprint early 05</td>
<td>58</td>
<td>60</td>
<td>63</td>
<td>1.3</td>
<td></td>
<td>58</td>
<td>60.3</td>
<td>64</td>
<td>1.7</td>
<td></td>
<td>56</td>
<td>57.1</td>
<td>60</td>
<td>1.3</td>
<td></td>
<td>56</td>
<td>59.5</td>
<td>65</td>
<td>2.0</td>
<td></td>
</tr>
<tr>
<td>sprint early 06</td>
<td>56</td>
<td>58.3</td>
<td>71</td>
<td>4.5</td>
<td></td>
<td>56</td>
<td>57.1</td>
<td>60</td>
<td>1.3</td>
<td></td>
<td>58</td>
<td>60.1</td>
<td>63</td>
<td>1.5</td>
<td></td>
<td>58</td>
<td>60.2</td>
<td>62</td>
<td>1.2</td>
<td></td>
</tr>
<tr>
<td>sprint early 07</td>
<td>59</td>
<td>60</td>
<td>63</td>
<td>1.2</td>
<td></td>
<td>59</td>
<td>60.1</td>
<td>63</td>
<td>1.5</td>
<td></td>
<td>57</td>
<td>58.6</td>
<td>60</td>
<td>1.0</td>
<td></td>
<td>57</td>
<td>60.6</td>
<td>61</td>
<td>1.1</td>
<td></td>
</tr>
<tr>
<td>sprint early 08</td>
<td>58</td>
<td>59.3</td>
<td>61</td>
<td>0.9</td>
<td></td>
<td>58</td>
<td>59.8</td>
<td>63</td>
<td>0.9</td>
<td></td>
<td>53</td>
<td>55.9</td>
<td>58</td>
<td>1.4</td>
<td></td>
<td>54</td>
<td>58.8</td>
<td>85</td>
<td>9.3</td>
<td></td>
</tr>
<tr>
<td>sprint early 09</td>
<td>59</td>
<td>59.3</td>
<td>61</td>
<td>0.7</td>
<td></td>
<td>59</td>
<td>60.2</td>
<td>62</td>
<td>1.2</td>
<td></td>
<td>53</td>
<td>55.9</td>
<td>58</td>
<td>1.4</td>
<td></td>
<td>54</td>
<td>58.8</td>
<td>85</td>
<td>9.3</td>
<td></td>
</tr>
<tr>
<td>sprint early 10</td>
<td>54</td>
<td>55.4</td>
<td>57</td>
<td>1.0</td>
<td></td>
<td>55</td>
<td>57.9</td>
<td>56</td>
<td>0.9</td>
<td></td>
<td>53</td>
<td>55.9</td>
<td>58</td>
<td>1.4</td>
<td></td>
<td>54</td>
<td>58.8</td>
<td>85</td>
<td>9.3</td>
<td></td>
</tr>
<tr>
<td>sprint hidden 01</td>
<td>44</td>
<td>105.1</td>
<td>148</td>
<td>35.8</td>
<td></td>
<td>47</td>
<td>102.4</td>
<td>141</td>
<td>35.4</td>
<td></td>
<td>50</td>
<td>108.3</td>
<td>145</td>
<td>27.4</td>
<td></td>
<td>40</td>
<td>83.3</td>
<td>128</td>
<td>34.7</td>
<td></td>
</tr>
<tr>
<td>sprint hidden 02</td>
<td>36</td>
<td>39.7</td>
<td>44</td>
<td>2.5</td>
<td></td>
<td>38</td>
<td>43.3</td>
<td>69</td>
<td>9.3</td>
<td></td>
<td>34</td>
<td>41.3</td>
<td>45</td>
<td>3.3</td>
<td></td>
<td>39</td>
<td>44.2</td>
<td>52</td>
<td>3.6</td>
<td></td>
</tr>
<tr>
<td>sprint hidden 03</td>
<td>69</td>
<td>83.4</td>
<td>111</td>
<td>16.0</td>
<td></td>
<td>70</td>
<td>97.6</td>
<td>140</td>
<td>27.1</td>
<td></td>
<td>69</td>
<td>98.1</td>
<td>137</td>
<td>29.4</td>
<td></td>
<td>68</td>
<td>91.8</td>
<td>151</td>
<td>28.7</td>
<td></td>
</tr>
<tr>
<td>sprint hidden 04</td>
<td>146</td>
<td>159.4</td>
<td>168</td>
<td>6.7</td>
<td></td>
<td>143</td>
<td>156.1</td>
<td>170</td>
<td>9.5</td>
<td></td>
<td>143</td>
<td>154.1</td>
<td>165</td>
<td>6.5</td>
<td></td>
<td>130</td>
<td>154.4</td>
<td>169</td>
<td>11.1</td>
<td></td>
</tr>
<tr>
<td>sprint hidden 05</td>
<td>66</td>
<td>88.7</td>
<td>143</td>
<td>27.6</td>
<td></td>
<td>62</td>
<td>85.2</td>
<td>151</td>
<td>30.3</td>
<td></td>
<td>65</td>
<td>77.2</td>
<td>106</td>
<td>16.0</td>
<td></td>
<td>68</td>
<td>88.8</td>
<td>149</td>
<td>30.5</td>
<td></td>
</tr>
<tr>
<td>sprint hidden 06</td>
<td>590</td>
<td>739.5</td>
<td>848</td>
<td>83.5</td>
<td></td>
<td>557</td>
<td>767.6</td>
<td>905</td>
<td>100.6</td>
<td></td>
<td>644</td>
<td>775.1</td>
<td>872</td>
<td>73.2</td>
<td></td>
<td>175</td>
<td>607.3</td>
<td>802</td>
<td>179.7</td>
<td></td>
</tr>
<tr>
<td>sprint hidden 07</td>
<td>213</td>
<td>341.2</td>
<td>437</td>
<td>72.5</td>
<td></td>
<td>251</td>
<td>384.1</td>
<td>498</td>
<td>71.4</td>
<td></td>
<td>192</td>
<td>379.5</td>
<td>588</td>
<td>101.0</td>
<td></td>
<td>235</td>
<td>370.5</td>
<td>441</td>
<td>67.7</td>
<td></td>
</tr>
<tr>
<td>sprint hidden 08</td>
<td>522</td>
<td>594.4</td>
<td>719</td>
<td>75.5</td>
<td></td>
<td>458</td>
<td>603.2</td>
<td>730</td>
<td>97.6</td>
<td></td>
<td>404</td>
<td>550.6</td>
<td>832</td>
<td>139.0</td>
<td></td>
<td>454</td>
<td>591.9</td>
<td>717</td>
<td>83.7</td>
<td></td>
</tr>
<tr>
<td>sprint hidden 09</td>
<td>885</td>
<td>933.9</td>
<td>981</td>
<td>30.8</td>
<td></td>
<td>897</td>
<td>961.9</td>
<td>1016</td>
<td>35.9</td>
<td></td>
<td>832</td>
<td>943.1</td>
<td>1005</td>
<td>52.1</td>
<td></td>
<td>804</td>
<td>873.1</td>
<td>907</td>
<td>38.3</td>
<td></td>
</tr>
<tr>
<td>sprint hidden 10</td>
<td>483</td>
<td>710.7</td>
<td>911</td>
<td>152.8</td>
<td></td>
<td>454</td>
<td>615.6</td>
<td>791</td>
<td>106.0</td>
<td></td>
<td>423</td>
<td>630</td>
<td>781</td>
<td>104.6</td>
<td></td>
<td>536</td>
<td>649.5</td>
<td>828</td>
<td>100.7</td>
<td></td>
</tr>
</tbody>
</table>

(Continued)
<table>
<thead>
<tr>
<th>Instance</th>
<th>max = 1.0</th>
<th>max = 1.3</th>
<th>max = 1.6</th>
<th>max = 1.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>sprint_hin01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sprint_hin02</td>
<td>170</td>
<td>307.9</td>
<td>389</td>
<td>66.9</td>
</tr>
<tr>
<td>sprint_hin03</td>
<td>186</td>
<td>333.1</td>
<td>446</td>
<td>78.9</td>
</tr>
<tr>
<td>medium_early01</td>
<td>436</td>
<td>448</td>
<td>458</td>
<td>7.3</td>
</tr>
<tr>
<td>medium_early02</td>
<td>430</td>
<td>446.5</td>
<td>453</td>
<td>6.7</td>
</tr>
<tr>
<td>medium_early03</td>
<td>432</td>
<td>441.4</td>
<td>450</td>
<td>5.1</td>
</tr>
<tr>
<td>medium_early04</td>
<td>427</td>
<td>449.6</td>
<td>458</td>
<td>9.0</td>
</tr>
<tr>
<td>medium_early05</td>
<td>501</td>
<td>509.4</td>
<td>518</td>
<td>6.6</td>
</tr>
<tr>
<td>medium_laden01</td>
<td>2513</td>
<td>2624.3</td>
<td>2735</td>
<td>64.9</td>
</tr>
<tr>
<td>medium_laden02</td>
<td>2019</td>
<td>2111.6</td>
<td>2205</td>
<td>55.5</td>
</tr>
<tr>
<td>medium_laden03</td>
<td>614</td>
<td>640.2</td>
<td>662</td>
<td>12.5</td>
</tr>
<tr>
<td>medium_laden04</td>
<td>616</td>
<td>629.9</td>
<td>644</td>
<td>9.1</td>
</tr>
<tr>
<td>medium_laden05</td>
<td>2395</td>
<td>2444.1</td>
<td>2513</td>
<td>44.8</td>
</tr>
<tr>
<td>medium_late01</td>
<td>1727</td>
<td>1828</td>
<td>1879</td>
<td>43.0</td>
</tr>
<tr>
<td>medium_late02</td>
<td>412</td>
<td>424.6</td>
<td>437</td>
<td>7.9</td>
</tr>
<tr>
<td>medium_late03</td>
<td>493</td>
<td>510.6</td>
<td>535</td>
<td>13.7</td>
</tr>
<tr>
<td>medium_late04</td>
<td>409</td>
<td>431.1</td>
<td>441</td>
<td>9.4</td>
</tr>
<tr>
<td>medium_late05</td>
<td>1787</td>
<td>1852.4</td>
<td>1913</td>
<td>37.2</td>
</tr>
<tr>
<td>medium_late06</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>medium_late07</td>
<td>561</td>
<td>589</td>
<td>618</td>
<td>17.6</td>
</tr>
<tr>
<td>medium_late08</td>
<td>2012</td>
<td>2071.4</td>
<td>2100</td>
<td>31.5</td>
</tr>
<tr>
<td>medium_late09</td>
<td>3742</td>
<td>4177.4</td>
<td>5042</td>
<td>359.4</td>
</tr>
</tbody>
</table>

Table 9. (Continued)
Table 9. (Continued)

<table>
<thead>
<tr>
<th>Instance</th>
<th>max = 1.0</th>
<th>max = 1.3</th>
<th>max = 1.6</th>
<th>max = 1.9</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>B.</td>
<td>M.</td>
<td>W.</td>
<td>S.</td>
</tr>
<tr>
<td>long.early01</td>
<td>502</td>
<td>51.2</td>
<td>52.4</td>
<td>7.4</td>
</tr>
<tr>
<td>long.early02</td>
<td>548</td>
<td>560.3</td>
<td>573</td>
<td>8.6</td>
</tr>
<tr>
<td>long.early03</td>
<td>475</td>
<td>506.8</td>
<td>524</td>
<td>15.2</td>
</tr>
<tr>
<td>long.early04</td>
<td>592</td>
<td>609.6</td>
<td>621</td>
<td>9.1</td>
</tr>
<tr>
<td>long.early05</td>
<td>577</td>
<td>589.5</td>
<td>598</td>
<td>6.4</td>
</tr>
<tr>
<td>long.hidden01</td>
<td>3987</td>
<td>4063.5</td>
<td>4167</td>
<td>49.4</td>
</tr>
<tr>
<td>long.hidden02</td>
<td>823</td>
<td>845.6</td>
<td>855</td>
<td>10.1</td>
</tr>
<tr>
<td>long.hidden03</td>
<td>765</td>
<td>779.9</td>
<td>794</td>
<td>10.0</td>
</tr>
<tr>
<td>long.hidden04</td>
<td>706</td>
<td>723.2</td>
<td>739</td>
<td>11.5</td>
</tr>
<tr>
<td>long.hidden05</td>
<td>859</td>
<td>899.4</td>
<td>941</td>
<td>22.7</td>
</tr>
<tr>
<td>long晚期01</td>
<td>3688</td>
<td>3766.9</td>
<td>3823</td>
<td>37.6</td>
</tr>
<tr>
<td>long晚期02</td>
<td>3717</td>
<td>3861.7</td>
<td>3990</td>
<td>74.3</td>
</tr>
<tr>
<td>long晚期03</td>
<td>3571</td>
<td>3673.2</td>
<td>3759</td>
<td>53.9</td>
</tr>
<tr>
<td>long晚期04</td>
<td>3625</td>
<td>3741.8</td>
<td>3848</td>
<td>83.1</td>
</tr>
<tr>
<td>long晚期05</td>
<td>3074</td>
<td>3149.1</td>
<td>3328</td>
<td>75.8</td>
</tr>
<tr>
<td>long乱01</td>
<td>767</td>
<td>785.1</td>
<td>807</td>
<td>12.1</td>
</tr>
<tr>
<td>long乱02</td>
<td>564</td>
<td>577.2</td>
<td>591</td>
<td>8.8</td>
</tr>
<tr>
<td>long乱03</td>
<td>3049</td>
<td>3151.5</td>
<td>3230</td>
<td>57.7</td>
</tr>
</tbody>
</table>

Harmony Search with Novel Selection Methods
Table 10. The behavior of harmony search-based methods using INRC2010 dataset.

<table>
<thead>
<tr>
<th>Instance</th>
<th>RMCS</th>
<th>GMCS</th>
<th>PMCS</th>
<th>TMCS</th>
<th>LMCS</th>
</tr>
</thead>
<tbody>
<tr>
<td>sprint_early01</td>
<td>58</td>
<td>59.4</td>
<td>61</td>
<td>1.1</td>
<td>60</td>
</tr>
<tr>
<td>sprint_early02</td>
<td>60</td>
<td>62.4</td>
<td>65</td>
<td>1.8</td>
<td>62</td>
</tr>
<tr>
<td>sprint_early03</td>
<td>53</td>
<td>56.8</td>
<td>58</td>
<td>1.7</td>
<td>53</td>
</tr>
<tr>
<td>sprint_early04</td>
<td>62</td>
<td>67.8</td>
<td>88</td>
<td>8.4</td>
<td>67</td>
</tr>
<tr>
<td>sprint_early05</td>
<td>59</td>
<td>62.1</td>
<td>75</td>
<td>4.7</td>
<td>60</td>
</tr>
<tr>
<td>sprint_early06</td>
<td>56</td>
<td>56.8</td>
<td>58</td>
<td>0.8</td>
<td>57</td>
</tr>
<tr>
<td>sprint_early07</td>
<td>57</td>
<td>62.5</td>
<td>73</td>
<td>5.3</td>
<td>60</td>
</tr>
<tr>
<td>sprint_early08</td>
<td>57</td>
<td>58.5</td>
<td>60</td>
<td>1.1</td>
<td>55</td>
</tr>
<tr>
<td>sprint_early09</td>
<td>57</td>
<td>62.5</td>
<td>74</td>
<td>5.9</td>
<td>58</td>
</tr>
<tr>
<td>sprint_early10</td>
<td>53</td>
<td>56.1</td>
<td>58</td>
<td>1.7</td>
<td>57</td>
</tr>
<tr>
<td>sprint_hidden01</td>
<td>41</td>
<td>87.3</td>
<td>118</td>
<td>31.6</td>
<td>51</td>
</tr>
<tr>
<td>sprint_hidden02</td>
<td>35</td>
<td>40.8</td>
<td>45</td>
<td>4.2</td>
<td>37</td>
</tr>
<tr>
<td>sprint_hidden03</td>
<td>70</td>
<td>92.5</td>
<td>138</td>
<td>26.5</td>
<td>78</td>
</tr>
<tr>
<td>sprint_hidden04</td>
<td>138</td>
<td>156.8</td>
<td>167</td>
<td>10.5</td>
<td>81</td>
</tr>
<tr>
<td>sprint_hidden05</td>
<td>62</td>
<td>83.2</td>
<td>123</td>
<td>23.2</td>
<td>74</td>
</tr>
<tr>
<td>sprint_hidden06</td>
<td>486</td>
<td>687</td>
<td>819</td>
<td>101.3</td>
<td>208</td>
</tr>
<tr>
<td>sprint_hidden07</td>
<td>286</td>
<td>436.3</td>
<td>637</td>
<td>108.7</td>
<td>211</td>
</tr>
<tr>
<td>sprint_hidden08</td>
<td>490</td>
<td>644.4</td>
<td>843</td>
<td>107.6</td>
<td>266</td>
</tr>
<tr>
<td>sprint_hidden09</td>
<td>874</td>
<td>945.2</td>
<td>976</td>
<td>33.9</td>
<td>395</td>
</tr>
<tr>
<td>sprint_hidden10</td>
<td>599</td>
<td>663.3</td>
<td>780</td>
<td>62.9</td>
<td>368</td>
</tr>
<tr>
<td>sprint_late01</td>
<td>45</td>
<td>58.4</td>
<td>99</td>
<td>20.1</td>
<td>52</td>
</tr>
<tr>
<td>sprint_late02</td>
<td>49</td>
<td>62.7</td>
<td>85</td>
<td>14.8</td>
<td>54</td>
</tr>
<tr>
<td>sprint_late03</td>
<td>56</td>
<td>76.9</td>
<td>115</td>
<td>21.7</td>
<td>59</td>
</tr>
<tr>
<td>sprint_late04</td>
<td>279</td>
<td>346.4</td>
<td>440</td>
<td>56.1</td>
<td>104</td>
</tr>
<tr>
<td>sprint_late05</td>
<td>51</td>
<td>57.7</td>
<td>80</td>
<td>10.6</td>
<td>59</td>
</tr>
<tr>
<td>sprint_late06</td>
<td>43</td>
<td>48.2</td>
<td>57</td>
<td>4.0</td>
<td>52</td>
</tr>
<tr>
<td>sprint_late07</td>
<td>68</td>
<td>117.9</td>
<td>172</td>
<td>33.8</td>
<td>64</td>
</tr>
<tr>
<td>sprint_late08</td>
<td>22</td>
<td>73.3</td>
<td>128</td>
<td>39.4</td>
<td>17</td>
</tr>
<tr>
<td>sprint_late09</td>
<td>24</td>
<td>63.3</td>
<td>127</td>
<td>40.0</td>
<td>17</td>
</tr>
<tr>
<td>sprint_late10</td>
<td>54</td>
<td>122.5</td>
<td>240</td>
<td>59.6</td>
<td>64</td>
</tr>
</tbody>
</table>
Table 10. (Continued)

<table>
<thead>
<tr>
<th>Instance</th>
<th>RMCS</th>
<th>GMCS</th>
<th>PMCS</th>
<th>TMCS</th>
<th>LMCS</th>
</tr>
</thead>
<tbody>
<tr>
<td>sprint_hint01</td>
<td>211</td>
<td>317.1</td>
<td>399</td>
<td>69.2</td>
<td>103</td>
</tr>
<tr>
<td>sprint_hint02</td>
<td>62</td>
<td>150</td>
<td>239</td>
<td>59.1</td>
<td>64</td>
</tr>
<tr>
<td>sprint_hint03</td>
<td>192</td>
<td>294.2</td>
<td>436</td>
<td>72.4</td>
<td>77</td>
</tr>
<tr>
<td>medium_early01</td>
<td>443</td>
<td>450.9</td>
<td>457</td>
<td>5.2</td>
<td>270</td>
</tr>
<tr>
<td>medium_early02</td>
<td>434</td>
<td>447.8</td>
<td>460</td>
<td>9.2</td>
<td>275</td>
</tr>
<tr>
<td>medium_early03</td>
<td>431</td>
<td>440.5</td>
<td>447</td>
<td>4.5</td>
<td>265</td>
</tr>
<tr>
<td>medium_early04</td>
<td>440</td>
<td>448.3</td>
<td>457</td>
<td>5.5</td>
<td>263</td>
</tr>
<tr>
<td>medium_early05</td>
<td>501</td>
<td>511.9</td>
<td>520</td>
<td>6.3</td>
<td>334</td>
</tr>
<tr>
<td>medium_hidden01</td>
<td>2440</td>
<td>2567.1</td>
<td>2688</td>
<td>76.0</td>
<td>253</td>
</tr>
<tr>
<td>medium_hidden02</td>
<td>2069</td>
<td>2127.7</td>
<td>2174</td>
<td>32.8</td>
<td>361</td>
</tr>
<tr>
<td>medium_hidden03</td>
<td>628</td>
<td>641.2</td>
<td>656</td>
<td>10.9</td>
<td>93</td>
</tr>
<tr>
<td>medium_hidden04</td>
<td>615</td>
<td>631.4</td>
<td>646</td>
<td>11.8</td>
<td>135</td>
</tr>
<tr>
<td>medium_hidden05</td>
<td>2313</td>
<td>2372.9</td>
<td>2466</td>
<td>44.3</td>
<td>280</td>
</tr>
<tr>
<td>medium_late01</td>
<td>1758</td>
<td>1802.5</td>
<td>1855</td>
<td>30.0</td>
<td>254</td>
</tr>
<tr>
<td>medium_late02</td>
<td>412</td>
<td>425.9</td>
<td>440</td>
<td>8.4</td>
<td>72</td>
</tr>
<tr>
<td>medium_late03</td>
<td>477</td>
<td>507.8</td>
<td>536</td>
<td>16.0</td>
<td>75</td>
</tr>
<tr>
<td>medium_late04</td>
<td>405</td>
<td>435.2</td>
<td>465</td>
<td>15.8</td>
<td>79</td>
</tr>
<tr>
<td>medium_late05</td>
<td>1746</td>
<td>1832.7</td>
<td>1922</td>
<td>47.7</td>
<td>238</td>
</tr>
<tr>
<td>medium_hint01</td>
<td>575</td>
<td>595.9</td>
<td>624</td>
<td>14.2</td>
<td>89</td>
</tr>
<tr>
<td>medium_hint02</td>
<td>1910</td>
<td>2043.9</td>
<td>2145</td>
<td>70.4</td>
<td>194</td>
</tr>
<tr>
<td>medium_hint03</td>
<td>3912</td>
<td>4190.1</td>
<td>4420</td>
<td>175.3</td>
<td>242</td>
</tr>
</tbody>
</table>

(Continued)
<table>
<thead>
<tr>
<th>Instance</th>
<th>RMCS</th>
<th>GMCS</th>
<th>PMCS</th>
<th>TMCS</th>
<th>LMCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>B.</td>
<td>M.</td>
<td>W.</td>
<td>S.</td>
<td>B.</td>
</tr>
<tr>
<td>early01</td>
<td>492</td>
<td>511.9</td>
<td>526</td>
<td>11.9</td>
<td>256</td>
</tr>
<tr>
<td>early02</td>
<td>550</td>
<td>561.8</td>
<td>576</td>
<td>8.3</td>
<td>299</td>
</tr>
<tr>
<td>early03</td>
<td>489</td>
<td>503.8</td>
<td>513</td>
<td>8.4</td>
<td>286</td>
</tr>
<tr>
<td>early04</td>
<td>587</td>
<td>61</td>
<td>623</td>
<td>11.6</td>
<td>356</td>
</tr>
<tr>
<td>early05</td>
<td>561</td>
<td>581.9</td>
<td>596</td>
<td>11.9</td>
<td>337</td>
</tr>
<tr>
<td>hidden01</td>
<td>3976</td>
<td>4128.2</td>
<td>4236</td>
<td>78.9</td>
<td>747</td>
</tr>
<tr>
<td>hidden02</td>
<td>818</td>
<td>846</td>
<td>872</td>
<td>18.3</td>
<td>225</td>
</tr>
<tr>
<td>hidden03</td>
<td>766</td>
<td>785.3</td>
<td>810</td>
<td>16.9</td>
<td>121</td>
</tr>
<tr>
<td>hidden04</td>
<td>719</td>
<td>738</td>
<td>759</td>
<td>12.1</td>
<td>134</td>
</tr>
<tr>
<td>hidden05</td>
<td>881</td>
<td>920.5</td>
<td>999</td>
<td>34.4</td>
<td>146</td>
</tr>
<tr>
<td>late01</td>
<td>3734</td>
<td>3825.6</td>
<td>4004</td>
<td>82.0</td>
<td>601</td>
</tr>
<tr>
<td>late02</td>
<td>3712</td>
<td>3883.2</td>
<td>3980</td>
<td>89.2</td>
<td>596</td>
</tr>
<tr>
<td>late03</td>
<td>3337</td>
<td>3671.1</td>
<td>3754</td>
<td>62.5</td>
<td>585</td>
</tr>
<tr>
<td>late04</td>
<td>3583</td>
<td>3752.8</td>
<td>3895</td>
<td>87.1</td>
<td>621</td>
</tr>
<tr>
<td>late05</td>
<td>3058</td>
<td>3188</td>
<td>3292</td>
<td>63.6</td>
<td>393</td>
</tr>
<tr>
<td>hint01</td>
<td>760</td>
<td>782.8</td>
<td>806</td>
<td>15.5</td>
<td>134</td>
</tr>
<tr>
<td>hint02</td>
<td>559</td>
<td>584.6</td>
<td>594</td>
<td>9.8</td>
<td>102</td>
</tr>
<tr>
<td>hint03</td>
<td>3036</td>
<td>3105.4</td>
<td>3160</td>
<td>44.3</td>
<td>375</td>
</tr>
</tbody>
</table>
Harmony Search with Novel Selection Methods

This is due to the fact that in the two selection methods, the new roster inherits the characteristics of the best roster in HM. Thus leads to the speeding up of the convergence for the proposed method.

It should be noted that the Global-best memory consideration selects the value of the current allocation of the new solution from the best solution in HM, if the feasibility is achieved. If not, in such case, it will select from the second best solution in HM, and so on until the last solution is reached. In other hand, the tournament memory consideration selects the value of the current allocation from the best solution in HM, if the new solution is feasible. If the feasibility is not achieved, then it will randomly choose from the current allocation feasible range.

Figure 1 plots the behavior of HSA using various memory consideration selection methods (i.e. RMCS, GMCS, PMCS, TMCS, and LMCS) when exploring the search space using some instances selected randomly from INRC2010 dataset. The run selected in this comparison was chosen randomly from the ten runs experimented with each memory consideration selection method for all instances of INRC2010 dataset. Note that, we use 5,000 iteration to show the differences among the different memory consideration selection methods visibly. The color lines in these diagrams show the correlation between the number of iterations and the objective function value (i.e. roster quality). An analysis of the diagrams shows that the roster quality improves as the number of iteration increases. The slope of the curves is relatively steep, which refers to the great improvement in the quality of rosters at the beginning of the search in all instances where there is possibly much of scope for improvement. The degree of improvement becomes relatively lower as the number of generations increases. Notably, the steepest slope curve that indicates which selection method is the best. Clearly, as shown in the figures, the TMCS method achieved the fastest rate of convergence and thus achieved best results in all instances. Also the GMCS method achieved the second fastest rate of convergence, while other selection methods obtained the worst rate of convergence.

Figure 2 shows the box plots that illustrate the distribution of roster quality of 10 runs for the various memory consideration selection methods using four instances selected randomly from INRC2010 dataset. The box plot represents the distribution of the results of 10 runs in visualization format. The x-axis represents the memory consideration selection methods (i.e. RMCS, GMCS, PMCS, TMCS, and LMCS), while the y-axis represents the objective function values. The box plot includes a red thick horizontal line within the box which represents the median, while the top and bottom of the box are the upper and lower quartiles. The dashed appendages refer to the spread and shape of the distribution of results of 10 runs. The upper pink line is meant for worst result (i.e. maximum), the lower green line is for the best result (i.e. minimum), and symbol “+” is used to represent outliers results of 10 runs. It can be seen that there is a small distance between the best, median, and worst results which demonstrates that the proposed method is stable. Visually, the GMCS achieved the best median of results for three out of four instances in this
Fig. 1. The various memory consideration selection methods for some instances of INRC2010 dataset.

figure. While the RMCS method obtained the best median of results for the fourth instance (i.e. sprint\textsubscript{late04}).

Figure 3 shows the distribution of the different results in HM for the different memory consideration selection methods at each iteration using long\_hidden05 instance. Note that, we use 100 iterations to show the distribution of the results visually. The color lines in these diagrams show the correlation between the number of iterations and the objective function values. These lines represent the best, mean, and worst results in HM at each iteration. An analysis of the diagram shows that the rate of the results distribution in HM decreases as the number of iteration
increases. Note that, the small distance among the best, mean, and worst lines that indicate small rate of distribution among the results in HM such as GMCS and TMCS. Clearly, the results of the Global-best memory consideration selection and the TMCS after 100 iterations are moving towards the same direction in HM. This is because the new roster inherited the characteristics of the best roster in HM and this led to the speeding up of the convergence. For the other memory consideration selection methods, the results are varied and this appears because of the big distance between the best, mean, and worst curves. This is due to the slow rate of convergence for these selection methods.

5.5. Comparison with others

The best results achieved by different harmony search-based methods with different memory consideration selection methods in memory consideration are compared with those produced by other methods. Brief summary for the other methods is as follow: Lü and Hao (2011) applied a multi start tabu search algorithm to tackle INRC2010 dataset. The solution method includes, a heuristic technique used to generate feasible solution, while using two neighborhood structures (i.e. move and swap) and three search strategies under a probability-based schema to improve the solution. Valouxis et al. (2012) used integer programming to tackle NRP. The solution method includes two phases for all tracks as follows: the first, assigning
Fig. 3. (Color online) The individuals distribution in HM of various memory consideration selection methods using long hidden 05 instance.

different nurses to working days while the second schedules the nurses assigned to working days and certain shifts. For medium and long tracks datasets, three additional neighborhood structures in the first phase were used. These neighborhoods include moving shifts between nurses, reschedule one specific day for all nurses, and reschedule two specific days for all nurses. Burke and Curtois (2010) tackled INRC2010 using two methods. The ejection chain-based method is used for the sprint track dataset while the branch and price method is used for medium and long tracks datasets. Bilgin et al. (2010) used a hyper-heuristic approach combined
Harmony Search with Novel Selection Methods

with a greedy shuffle move. The hyper-heuristic was initially used to generate a feasible roster. The greedy shuffle was used in the improvement loop by swaps of partial rosters between nurses. Finally, Nonobe (2010) reformulate the problem as Constraint Optimization Problem (COP), and then used the “COP solver” based on tabu search to tackle INRC2010 dataset.

Table 11 includes the best results of the our harmony search-based methods and the other comparative methods for tracks sprint, medium, and long respectively.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>sprint_early01</td>
<td>58</td>
<td>56</td>
<td>56</td>
<td>56</td>
<td>57</td>
<td>56</td>
</tr>
<tr>
<td>sprint_early02</td>
<td>58</td>
<td>58</td>
<td>58</td>
<td>58</td>
<td>59</td>
<td>58</td>
</tr>
<tr>
<td>sprint_early03</td>
<td>53</td>
<td>51</td>
<td>51</td>
<td>51</td>
<td>51</td>
<td>51</td>
</tr>
<tr>
<td>sprint_early04</td>
<td>62</td>
<td>59</td>
<td>59</td>
<td>59</td>
<td>60</td>
<td>59</td>
</tr>
<tr>
<td>sprint_early05</td>
<td>58</td>
<td>58</td>
<td>58</td>
<td>58</td>
<td>58</td>
<td>58</td>
</tr>
<tr>
<td>sprint_early06</td>
<td>55</td>
<td>54</td>
<td>54</td>
<td>54</td>
<td>54</td>
<td>54</td>
</tr>
<tr>
<td>sprint_early07</td>
<td>57</td>
<td>56</td>
<td>56</td>
<td>56</td>
<td>56</td>
<td>56</td>
</tr>
<tr>
<td>sprint_early08</td>
<td>57</td>
<td>56</td>
<td>56</td>
<td>56</td>
<td>56</td>
<td>56</td>
</tr>
<tr>
<td>sprint_early09</td>
<td>57</td>
<td>55</td>
<td>55</td>
<td>55</td>
<td>55</td>
<td>55</td>
</tr>
<tr>
<td>sprint_early10</td>
<td>53</td>
<td>52</td>
<td>52</td>
<td>52</td>
<td>52</td>
<td>52</td>
</tr>
<tr>
<td>sprint_hidden01</td>
<td>40</td>
<td>32</td>
<td>33</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>sprint_hidden02</td>
<td>34</td>
<td>32</td>
<td>32</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>sprint_hidden03</td>
<td>68</td>
<td>62</td>
<td>62</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>sprint_hidden04</td>
<td>75</td>
<td>66</td>
<td>67</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>sprint_hidden05</td>
<td>62</td>
<td>59</td>
<td>59</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>sprint_hidden06</td>
<td>174</td>
<td>130</td>
<td>134</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>sprint_hidden07</td>
<td>167</td>
<td>153</td>
<td>153</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>sprint_hidden08</td>
<td>244</td>
<td>204</td>
<td>209</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>sprint_hidden09</td>
<td>352</td>
<td>338</td>
<td>338</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>sprint_hidden10</td>
<td>348</td>
<td>306</td>
<td>306</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>sprint_late01</td>
<td>42</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>40</td>
<td>37</td>
</tr>
<tr>
<td>sprint_late02</td>
<td>46</td>
<td>42</td>
<td>42</td>
<td>42</td>
<td>44</td>
<td>42</td>
</tr>
<tr>
<td>sprint_late03</td>
<td>54</td>
<td>48</td>
<td>48</td>
<td>48</td>
<td>50</td>
<td>48</td>
</tr>
<tr>
<td>sprint_late04</td>
<td>99</td>
<td>73</td>
<td>75</td>
<td>73</td>
<td>81</td>
<td>76</td>
</tr>
<tr>
<td>sprint_late05</td>
<td>48</td>
<td>44</td>
<td>44</td>
<td>44</td>
<td>45</td>
<td>45</td>
</tr>
<tr>
<td>sprint_late06</td>
<td>43</td>
<td>42</td>
<td>42</td>
<td>42</td>
<td>42</td>
<td>42</td>
</tr>
<tr>
<td>sprint_late07</td>
<td>48</td>
<td>42</td>
<td>42</td>
<td>42</td>
<td>46</td>
<td>43</td>
</tr>
<tr>
<td>sprint_late08</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>sprint_late09</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>sprint_late10</td>
<td>54</td>
<td>43</td>
<td>43</td>
<td>43</td>
<td>46</td>
<td>44</td>
</tr>
<tr>
<td>sprint_hint01</td>
<td>91</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>78</td>
<td>—</td>
</tr>
<tr>
<td>sprint_hint02</td>
<td>53</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>47</td>
<td>—</td>
</tr>
<tr>
<td>sprint_hint03</td>
<td>75</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>57</td>
<td>—</td>
</tr>
<tr>
<td>medium_early01</td>
<td>263</td>
<td>240</td>
<td>240</td>
<td>240</td>
<td>242</td>
<td>241</td>
</tr>
<tr>
<td>medium_early02</td>
<td>275</td>
<td>240</td>
<td>240</td>
<td>240</td>
<td>241</td>
<td>240</td>
</tr>
<tr>
<td>medium_early03</td>
<td>259</td>
<td>236</td>
<td>236</td>
<td>236</td>
<td>238</td>
<td>236</td>
</tr>
<tr>
<td>medium_early04</td>
<td>257</td>
<td>237</td>
<td>237</td>
<td>237</td>
<td>238</td>
<td>238</td>
</tr>
<tr>
<td>medium_early05</td>
<td>327</td>
<td>303</td>
<td>303</td>
<td>303</td>
<td>304</td>
<td>304</td>
</tr>
</tbody>
</table>

(Continued)
Table 11. (Continued)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>medium_hidden01</td>
<td>235</td>
<td>117</td>
<td>130</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>medium_hidden02</td>
<td>345</td>
<td>220</td>
<td>221</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>medium_hidden03</td>
<td>80</td>
<td>35</td>
<td>36</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>medium_hidden04</td>
<td>127</td>
<td>79</td>
<td>81</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>medium_hidden05</td>
<td>269</td>
<td>119</td>
<td>122</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>medium late01</td>
<td>254</td>
<td>164</td>
<td>158</td>
<td>157</td>
<td>163</td>
<td>176</td>
</tr>
<tr>
<td>medium late02</td>
<td>72</td>
<td>20</td>
<td>18</td>
<td>18</td>
<td>21</td>
<td>29</td>
</tr>
<tr>
<td>medium late03</td>
<td>68</td>
<td>30</td>
<td>29</td>
<td>29</td>
<td>32</td>
<td>30</td>
</tr>
<tr>
<td>medium late04</td>
<td>78</td>
<td>36</td>
<td>35</td>
<td>35</td>
<td>38</td>
<td>37</td>
</tr>
<tr>
<td>medium late05</td>
<td>238</td>
<td>117</td>
<td>107</td>
<td>107</td>
<td>122</td>
<td>125</td>
</tr>
<tr>
<td>medium hint01</td>
<td>83</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>40</td>
<td>—</td>
</tr>
<tr>
<td>medium hint02</td>
<td>191</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>84</td>
<td>—</td>
</tr>
<tr>
<td>medium hint03</td>
<td>219</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>129</td>
<td>—</td>
</tr>
<tr>
<td>long early01</td>
<td>256</td>
<td>197</td>
<td>197</td>
<td>197</td>
<td>197</td>
<td>197</td>
</tr>
<tr>
<td>long early02</td>
<td>299</td>
<td>222</td>
<td>219</td>
<td>219</td>
<td>220</td>
<td>219</td>
</tr>
<tr>
<td>long early03</td>
<td>286</td>
<td>240</td>
<td>240</td>
<td>240</td>
<td>240</td>
<td>240</td>
</tr>
<tr>
<td>long early04</td>
<td>356</td>
<td>303</td>
<td>303</td>
<td>303</td>
<td>303</td>
<td>303</td>
</tr>
<tr>
<td>long early05</td>
<td>337</td>
<td>284</td>
<td>284</td>
<td>284</td>
<td>284</td>
<td>284</td>
</tr>
<tr>
<td>long hidden01</td>
<td>364</td>
<td>346</td>
<td>363</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>long hidden02</td>
<td>225</td>
<td>89</td>
<td>90</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>long hidden03</td>
<td>97</td>
<td>38</td>
<td>38</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>long hidden04</td>
<td>102</td>
<td>22</td>
<td>22</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>long hidden05</td>
<td>118</td>
<td>45</td>
<td>41</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>long late01</td>
<td>353</td>
<td>237</td>
<td>235</td>
<td>235</td>
<td>241</td>
<td>235</td>
</tr>
<tr>
<td>long late02</td>
<td>511</td>
<td>229</td>
<td>229</td>
<td>229</td>
<td>245</td>
<td>229</td>
</tr>
<tr>
<td>long late03</td>
<td>559</td>
<td>222</td>
<td>220</td>
<td>220</td>
<td>233</td>
<td>220</td>
</tr>
<tr>
<td>long late04</td>
<td>453</td>
<td>227</td>
<td>221</td>
<td>221</td>
<td>246</td>
<td>221</td>
</tr>
<tr>
<td>long late05</td>
<td>390</td>
<td>83</td>
<td>83</td>
<td>83</td>
<td>87</td>
<td>83</td>
</tr>
<tr>
<td>long hint01</td>
<td>134</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>33</td>
<td>—</td>
</tr>
<tr>
<td>long hint02</td>
<td>102</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>17</td>
<td>—</td>
</tr>
<tr>
<td>long hint03</td>
<td>375</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>55</td>
<td>—</td>
</tr>
</tbody>
</table>

It should be noted that the best results among different methods are highlighted in bold (i.e. lowest is the best). Apparently, the harmony search-based methods obtained the best published results in four instances, and comparable results for the other instances of INRC2010 dataset.

6. Conclusion

The selection methods of a population-based algorithms provides the driving force to generate good-enough solutions based on natural rule of Darwin’s principle survival-of-the-fittest. Iteratively, the selection methods select the individuals with high fitness to be as members of the population of next iteration. The HSA is a population-based metaheuristic proposed by Geem et al. (2001). It improves the new solution iteratively using three operators: (i) memory consideration; (ii) random
Harmony Search with Novel Selection Methods

consideration; (iii) pitch adjustment. In this paper, we replaced the original random selection of the memory consideration operator with a set of selection methods in order to speed-up the convergence. These selection methods include the tournament, proportional, linear rank, and Global-best selection. In memory consideration, the selection methods select the individuals with high fitness to generate new harmony to hope to get better results. The proposed HSA variations (i.e. five HSA variations are proposed where each of which is HSA with one selection method in memory consideration) are evaluated using standard dataset published by the first International Nurse Rostering Competition INRC2010. NRP is a combinatorial optimization problem tackled by assigning a set of qualified nurses to a set of shifts over a predefined scheduling period. Experimentally, the TMCS method achieved the best rate of convergence and the best results in comparison with other memory consideration selection methods. This is because the new solution inherits the characteristics of the best solution in the population and this leads to the speeding up of the convergence. Our future directions involve

- Integrate our method with local-based metaheuristics to enhance the exploitation capability.
- Investigate other neighborhood structures in pitch adjustment operator.
- Relaxation for some or all of the hard constraints, hoping to get better results.
- Tune different parameters of HSA for NRP.

References


M. A. Awadallah et al.


Mohammed A. Awadallah received his BSc degree in Computer Science from Islamic University of Gaza, Palestine, and MSc in Computer Information System (CIS) from Arab Academy for Banking and Financial Sciences, Amman, Jordan in 2002 and 2005, respectively. Since 2005, he works as a lecturer in the Department of Computer Science, College of Science, Al-Aqsa University-Gaza, Palestine. He is currently toward achieving the PhD from the School of Computer Sciences, Universiti Sains Malaysia. His current research interests include evolutionary algorithms, nature-inspired computing, and their applications to timetabling problems etc.

Ahmad Tajudin Khader received his BSc and MSc degrees in Mathematics from the University of Ohio, USA in 1982 and 1983, respectively. He obtained his PhD in Computer Science from the University of Strathclyde, U.K. in 1993. He is currently working as a professor and Dean in the School of Computer Sciences, Universiti Sains Malaysia. He has authored and co-authored hundreds of high quality papers in international journals, conferences, and book.

He is a member of the Institute of Electrical and Electronics Engineers (IEEE) as an international professor, Association of Computing Machinery (ACM), Special Interest Group for Genetic and Evolutionary Computation (SIGEVO), and Computational Intelligence Society. His research interests mainly focus on optimization and scheduling.

Mohammed Azmi Al-Betar is an assistant professor in the Department of Computer Science at Al-Huson University College, Al-Balqa Applied University, Irbid, Jordan and a senior member of the computational intelligence research group in
the school of computer science at University Sains Malaysia (USM). He received his BSc and MSc degrees from Computer Science Department at Yarmouk University, Irbid, Jordan in 2001 and 2003 respectively.

He obtained his PhD from the School of Computer Sciences at the University Sains Malaysia (USM), Pulau Penang, Malaysia in October 2010. The author published a number of high quality papers in international journals and conferences. His research interests are mainly directed to metaheuristic optimization methods and hard combinatorial optimization problems including scheduling and timetabling.

Asaju La’aro Bolaji received his BTech degree in Physics from Federal University of Technology, Minna, Nigeria and MSc in Mathematics from the University of Ilorin, Nigeria in 2001 and 2006, respectively. He is a member of the Academic staff in the Department of Computer Science, University of Ilorin, Ilorin, Nigeria and currently working toward obtaining PhD from the School of Computer Sciences, Universiti Sains Malaysia. His research interests include evolutionary algorithms, nature-inspired computation, and their applications to optimization problems etc.