ICIT 2015 The 7th International Conference on Information Technology
doi:10.15849/icit.2015.0005

Solving Nurse Rostering Problem Using Artificial Bee Colony Algorithm

Asaju, La'aro Bolaji
Department of Computer Science, University of Ilorin, P.M.B. 1515 Ilorin, Nigeria.
lbasaju@unilorin.edu.ng

Mohammed A. Awadallah
Department of Computer Science, Al-Aqsa University, P. O. Box 4051, Gaza, Palestine.
ma.awadallah@alaqsa.edu.ps

Mohammed Azmi Al-Betar
Department of Information Technology, Al-Huson University College, Al-Balqa Applied University, P. O. Box 50, Al-Huson, Irbid, Jordan.
mohbetar@bau.edu.jo

Ahamad Tajudin Khader
School of Computer Sciences, Universiti Sains Malaysia, Penang, Malaysia.
tajudin@cs.usm.my

***Abstract*—Artificial bee colony algorithm(ABC) is proposed as a new nature-inspired algorithm which has been successfully utilized to tackle numerous class of optimization problems belongs to the category of swarm intelligence optimization algorithms. The major focus of this paper is to show that ABC could be used to generate good solutions when adapted to tackle the nurse rostering problem (NRP). In the proposed ABC for the NRP, the solution methods is divided into two phases. The first uses a heuristic ordering strategy to generate feasible solutions while the second phase employs the usage of ABC algorithm in which its operators are utilized to enhance the feasible solutions to their optimality. The proposed algorithm is tested on a set of 69 problem instances of the dataset introduced by the First International Nurse Rostering Competition 2010 (INRC2010). The results produced by the proposed algorithm are very promising when compared with some existing techniques that worked on the same dataset. Further investigation is still necessary for further improvement of the proposed algorithm.**

Keywords—***Nurse Rostering; Artificial Bee Colony Algorithm; Swarm Intelligence Method; Nature-inspired algorithm***

I. INTRODUCTION

The nurse rostering problem (NRP) is among the timetabling problem that is widely investigated by the researchers in the domain of operations research and artificial intelligence. The NRP as a NP-hard problem is described as an assignment of a set of qualified nurses to a different set of shifts over a predetermined scheduling period, subject to a set of hard and soft constraints. The hard constraints is the type that must be fulfilled for the roster to be *feasible* whereas the violations of soft constraints in the NRP are allowed but should be minimized as much as possible. It is noteworthy that the quality of the roster is determined by the satisfaction of the soft constraints in a feasible roster. The basic objective of NRP is to generate a feasible roster of high quality. However, studies in NRP domain have shown that it is almost impossible to find a roster that satisfies all constraints, since the NRP is classified as a combinatorial optimization problem [1] [2]. Due to the combinatorial and highly constrained nature of NRP, providing good quality roster is a very difficult and challenging task [3]. Naturally, investigations of numerous techniques for tackling NRP in the timetabling domain have increased over the past five decades. Some earliest techniques utilized for the NRP include integer programming [4], [5], goal programming [6], case-based reasoning [7], [8] and constraint Programming [9], [10]. In the recent time, some of the metaheuristic techniques that have also been employed for the NRP are local search-based approaches, which include tabu search [11], [12], simulated annealing [13], variable neighbourhood structures (VNS) [14], [15]. Others are population-based approaches like ant colony optimization [16], genetic algorithm (GA) [17], [18], harmony search algorithm (HSA) [19], [20], particle swarm optimization [21]. Similarly, hyperheuristic and hybrid metaheuristic approaches have also been utilized [22]. The comprehensive review of the methodologies utilized in tackling NRP can be found in [23], [24], [25].

This paper tackles the NRP dataset proposed by the First International Nurse Rostering Competition (INRC2010), which is organized by the CODeS research group at Katholieke Universiteit Leuven in Belgium, SINTEF Group in Norway and the University of Udine in Italy. The dataset of the INRC2010 is classified into three tracks: sprint, medium, and long datasets, which are varied in size and complexity. Each track is grouped into four categories in accordance with their publication time at the competition: early, late, hidden, and hint. Few techniques proposed to solve the INRC2010 dataset during and after the competition are review as follows.

Valouxis et al. in [26] applied Integer Programming (IP) to tackle the NRP using INRC2010 dataset in which their solution method consists of two stages: the first stage consists of assigning different nurses to working days whereas the second stage involves scheduling of the nurses assigned to working days to certain shifts. The authors employed the use of three additional neighborhood structures in the first phase which are: (i) rescheduling one day in the roster for another time, (ii) rescheduling two days in the roster for another time, and (iii) reshuffling the shifts among nurses, for the medium and long track of the dataset.

doi:10.15849/icit.2015.0005 © ICIT 2015 (http://icit.zuj.edu.jo/ICIT15)

The method ranked first in all three tacks of the dataset. The presentation of two methods to solve the INRC2010 dataset is presented in [27]. The authors in their work utilized the ejection chain-based method for the sprint track dataset while the branch and price method is employed for medium and long tracks of the INRC2010 dataset. The branch and price method achieved second rank for the medium and long tracks while the ejection method came fourth in sprint track of the dataset. The modeling of INCR2010 dataset as Constraint Optimization Problem (COP) is given in [28]. The author utilized the "COP solver" based on tabu search to further enhanced the results and the technique rated second, third, fourth in sprint, medium and long tracks of INRC2010, respectively.

Application of adaptive local search based on tabu search to tackle INRC2010 dataset is presented in [29] in which the solution method is also divided into two stages. The first phase involves the use of a random heuristic method to generates a feasible roster while the utilization of two neighborhood structures (i.e., move and swap) were employed to improve the solution at the second stage. It is worthy to mention that the method maintained the previous rosters in an elite pool. If the quality of the roster could not be improved within a given number of iterations with the aid of local search procedure, then one of the elite rosters is randomly chosen to restarts the second stage. The method achieved third and fourth position in the sprint and medium tracks respectively. The hybridization of a hyper-heuristic with a greedy shuffle move to solve INRC2010 dataset is presented by Bilgin et al. [30]. At initial stage, simulated annealing hyper-heuristic was employed to generate a feasible roster, where the satisfaction of soft constraints is achieved as much as possible. The greedy shuffle was used for further improvement of the roster. The hybrid hyper-heuristic method came third in long track, and fifth in sprint and medium tracks of the INRC2010 dataset. The introduction of a heuristic method for solving the INRC2010 dataset is considered in [31]. The heuristic method was employed in the construction of a feasible roster as well as trying to achieved the satisfaction of five pre-defined soft constraints. The authors utilized three local search procedures to further enhanced the roster. The method achieved the fifth position in long track. Adaptation of harmony search algorithm (HSA) was proposed for NRP using INRC2010 dataset in [19] where the results achieved on small instances of the dataset shows that method is very promising. The HSA was later modified in another development with inclusion of specific local search procedures in the pitch adjustment operator to minimized the violations of the soft constraints in [19]. The performance of modified HSA is further enhanced with the hybridization of greedy shuffle local search procedure which was utilized to enhance the new solution locally at each iteration [33]. Other HSA related works that have been employed to tackle NRP can be found in [20], [34], [35]. It is worth noting that research in the domain of NRP is still active, since exact solution has as yet been found for the INRC2010 dataset which necessitated further investigations using other algorithmic techniques. The main purpose of this study is investigated whether the use of the Artificial Bee Colony Algorithm (ABC) could be utilized to improve the state-of-the-art results for the INRC2010 dataset.

II. NURSE ROSTERING PROBLEM

The Nurse Rostering Problem (NRP) could be solved by assigning a set of nurses with different skills and work contracts to a set of shift types over a given scheduling period. The solution (or roster) to NRP is subject to hard and soft constraints. The hard constraints in NRP must be fulfilled in the roster (i.e. H_1 and H_2 as shown in Table I). The fulfillment of soft constraints (i.e. S_1- S_{10} see Table I) is desirable, and determines the quality of the roster. The basic objective is to find a roster that satisfies all hard constraints while minimizing soft constraints' violations. Table I shows the hard and soft constraints of INRC2010 datasets.

TABLE I
INRC2010 HARD AND SOFT CONSTRAINTS.

Hard Constraints	
H1	All demanded shifts must be assigned to a nurse.
H2	A nurse can only work one shift per day, i.e., no two shifts can be assigned to the same nurse on a day.
Soft Constraints	
S1	Maximum and minimum number of assignments for each nurse during the scheduling period.
S2	Maximum and minimum number of consecutive working days.
S3	Maximum and minimum number of consecutive free days.
S4	Assign complete weekends.
S5	Assign identical complete weekends.
S6	Two free days after a night shift.
S7	Requested day-on/off.
S8	Requested shift-on/off.
S9	Alternative skill.
S10	Unwanted patterns. (Where a pattern is a set of legal shifts defined in terms of work to be done during the shifts; Wren, 1996.)

Mathematically, the hard constraints for the INRC2010 dataset can be formulated as follows:

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

$$\sum_{i=1}^{N} x_i = d_{jk}. \quad (1)$$

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

$$\sum_{i=1}^{N} x_i \leq 1. \quad (2)$$

doi:10.15849/icit.2015.0005

Note that x_i is the allocation in the nurse roster (i.e solution) ***(x)*** assigned with a three elements (nurse *u*, day *v*, shift *r*). d_{jk} is the number of nurses required for day *(j)* at shift *(k)*, where $v = j$, $r = k$, and *N* represents the maximum length of allocations for nurse roster ***(x)*** as calculated in Eq. (3)

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

where *W* represents the total number of weeks in a scheduling period, *T* represents the total number of shifts.

The nurse solution (i.e. roster) is evaluated using an objective cost in Eq. (4), which sums up the penalty of soft constraint violations in a feasible roster.

$$min\ \boldsymbol{f(x)} \sum_{s=1}^{10} \boldsymbol{c_s \bullet g_s(x)} \qquad (4)$$

It is noteworthy that s is the index of the soft constraint $(S_1, \ldots\ldots, S_{10})$, c_s is the penalty weight for the violation of the soft constraint s, while $g_s(x)$ represent the total number of violations in ***x*** for the soft constraint *s*, ***x*** is a roster solution which represented as a vector shown in Fig 1

Fig. 1, Roster ***x*** representation

x_1	x_2	x_3	....	x_{N-1}	x_N
Nurse 2 Day 5 Shift L	Nurse 5 Day 10 Shift E	Nurse 1 Day 9 Shift N	....	Nurse 11 Day 1 Shift D	Nurse 3 Day 19 Shift D

III. ARTIFICIAL BEE COLONY ALGORITHM

Artificial Bee Colony Algorithm (ABC) is a nature-inspired stochastic algorithm developed in 2005 by Karaboga in [36] for solving numerical problems. The ABC algorithm as a population-based search method is motivated by intelligent foraging behaviour of honey bee in their hives based on the model proposed in [37]. In ABC, the colony comprising three classes of bees: employed, onlooker, and scout bees. The colony is divided into two where the first half is occupies by the employed bees, while the second half consists of the onlookers bees. Each employed bee is associated with a solution (i.e., food source). In other words, the number of employed bees is equal to the number of food sources. The employed bee whose food source is abandoned by the onlooker automatically turns to a scout. The onlooker bees are those bees that hang around the hive to understudy the dance behavior of the employed bees in order to choose the desired solution. The scouts are ones that are randomly exploring the solution search space for new food sources. Normally, the number of food sources in ABC algorithm is equal to the number of solutions in the population. Furthermore, the position of a food source signifies the possible solution for the optimization problem. The nectar amount of a solution (i.e., food source) represents the quality of the food source by that solution [36].

ABC algorithm has been applied and hybridized successfully for tackling real-world problems especially numerous formulations of the timetabling problems [38], [39]. A comprehensive review for ABC algorithm applications on several combinatorial optimization problems can be found in [40], [41].

A. Artificial Bee Colony for NRP

In this section, the concepts of Artificial Bee Colony Algorithm (ABC) as adapted for the NRP is discussed. The adapted ABC algorithm involves changing its continuous nature with integration of different neighourbood structures in order to cope with the solution search space of NRP.

The nurse roster (i.e. solution) is represented as a vector of allocations $x = (x_1, x_2, \ldots, x_N)$ where each allocation contains three values (nurse, day, shift). For instance, let ***x*** *= (1, 1, 1); (1, 2, 3),..., (5, 5, 4)* be a feasible nurse roster. The roster is interpreted by ABC algorithm as follows: the allocation x_1 = *(1, 1, 1)* means nurse n_1 is assigned to shift s_1 at day d_1. The second allocation x_2 = *(0, 1, 2)* means to nurse n_0 assigned to shift s_2 at day d_1, and so on. Note that representation of this roster is adopted in [20]. The description of six main procedural steps of ABC as algorithm adapted for tackling NRP are given as follows:

1) Initialization of ABC and INRC2010 parameters: This step involves initialization of the three control parameters of adapted ABC that are needed for tackling the NRP: solution number *(SN)* which is the number of food sources in the population and similar to the population size in GA; maximum cycle number *(MCN)* which represents the maximum number of iterations; and limit that is responsible for the abandonment of solution, if there is no improvement for certain number of iterations and basically use in diversifying the search. Similarly, the NRP parameters that are drawn from the *INRC2010* dataset are also initialized. They are the set of nurses, the set of skill categories, the set of shift types, the scheduling period, the set of work contracts, matrix of weekly nurse demand, matrices of nurses preferences, and eventually the set of unwanted patterns. The job specification which includes: total number of shifts, minimum number of shifts, maximum number of consecutive working days, minimum number of consecutive working days, maximum number of consecutive free days, minimum number of consecutive free days, and maximum working weekend in four weeks.

2) Initialization of the Food Source Memory: The food source memory *(FSM)* is a memory allocation that consists

doi:10.15849/icit.2015.0005 © ICIT 2015 (http://icit.zuj.edu.jo/ICIT15)

of sets of feasible food source (i.e. rosters) which is determined by *SN* as shown in Eq. 5 In this step, the feasible rosters are generated using the heuristic ordering approach and stored in ascending order in FSM according to the objective cost values that is $f(\boldsymbol{x}_1)$, $f(\boldsymbol{x}_2),\ldots,f(\boldsymbol{x}_{SN})$. The function of heuristic ordering is to sorts the daily shifts in ascending order based on the level of difficulty. It noteworthy that the lowest weekly nurses demand is the most difficulty and thus, the required nurses of the ordered shifts will be scheduled starting with the most difficult and ending with less difficult one.

$$\textbf{FSM} = \begin{bmatrix} x_1(1) & x_1(2) & \cdots & x_1(N) \\ x_2(1) & x_2(2) & \cdots & x_2(N) \\ \vdots & \vdots & \ddots & \vdots \\ x_{SN}(1) & x_{SN}(2) & \cdots & x_{SN}(N) \end{bmatrix} \begin{bmatrix} f(\boldsymbol{x}_1) \\ f(\boldsymbol{x}_2) \\ \vdots \\ f(\boldsymbol{x}_{SN}) \end{bmatrix} \qquad (5)$$

3) *Send the Employed Bee to the Food Sources:* in this step, the employed bee operator selects feasible nurse rosters sequentially from the FSM and exploits each roster using the neighbourhood structures to produce a new set of neighbouring solutions. The neighbourhood structures utilized by employed bee are:

- **Move Neighbourhood Structure (MNS):** The nurse of chosen allocation x_j is replaced with another nurse selected randomly to solve the violations of the soft constraint.
- **Swap Neighbourhood Structure (SNS):** The shift of selected allocation x_j is swapped with another shift on the same day for another selected allocation x_k.
- **Swap Unwanted Pattern (SUP):** This exchange a group of shifts among two nurses in which the chosen allocation x_j is replaced with another group of shifts on the same day for another chosen allocation x_k.
- **Token Ring Move (TRM).** The nurse of chosen allocation x_j is replaced by another nurse selected randomly, if the soft constraint S_7 is violated. Furthermore, the shift of a selected allocation x_j will be exchanged with another shift on which another nurse is working on the same day, for another selected allocation x_k to solve the violation of the soft constraint S_8.

The fitness of each new roster is calculated. If it is better than that of candidate roster (i.e. food source), then it replaces the parent roster in FSM. This process is implemented for all solutions. The detailed of this process is given in Algorithm (1).

4) *Send the Onlooker Bees to the Food Sources:* Subsequent to the completion of employed bees exploitation process, the employed bees share the information of exploited food source (i.e. roster) with onlooker bees. The onlooker bees decide to follow certain employed bees and exploit their corresponding food sources randomly using the set of neighbourhood structures discussed above based on proportional selection probability as shown in Eq. (6)

$$p_j = \frac{f(\boldsymbol{x}_j)}{\sum_{k=1}^{SN} f(\boldsymbol{x}_k)}$$

```
for i = 1 ··· SN  do
  i = RND() {RND generates a random integer number
  in range 1 - 4}
  if i = 1 then
    x^{i(new)} = MNS(x^i)
  else
    if i = 2 then
      x^{i(new)} = SNS(x^i)
    else
      if i = 3 then
        x^{i(new)} = SUP(x^i)
      else
        if i = 4 then
          x^{i(new)} = TRM(x^i)
        end if
      end if
    end if
  end if
  if x^{i(new)} is better than x^i then
    x^i = x^{i(new)}
  end if
  next i
end for
```

Algorithm 1: Employed Bee Phase

Note that the $\sum_{i=1}^{SN} p_i$ is unity

Thus, the roster with higher selection probability may be selected and adjusted to its neighbourhood using the same strategy as the employed bee. The fitness of the new roster is calculated and if it is better, then it replaces the current one.

5) *Send the Scout to Search for Possible New Food Sources:* Owing to continuous exploitation, some food sources may finally be exhausted in which they might be abandoned by its corresponding employed bee. Thus, the associated employed bee turns to a scout bee, and explores the solution search space randomly for a possible new food source to replace the abandoned one. Memorize the fitness of the best food source found so far in FSM.

6) *Stopping condition:* Repeat steps 3-5 until a stop condition is achieved, which is originally determined by *MCN*.

IV. Experimental Results and Analysis

In this section, the proposed adapted ABC for NRP is coded in Microsoft Visual C++ 6.0 on Windows 7 platform

doi:10.15849/icit.2015.0005 © ICIT 2015 (http://icit.zuj.edu.jo/ICIT15)

on Intel 2.00 GHz Core 2.66 Quad processor with 2 GB of RAM. A dataset introduced by INRC2010 for nurse rostering is employed to evaluates the performance of the proposed ABC for NRP. The dataset is grouped into three tracks: sprint, medium, and long problem instances based on complexity and size. Each track of the competition is categorized into four types according to the publication time with reference to the competition: early, late, hidden, and hint. The sprint track comes with 33 problem instances that are classified into 10: early, 10: late, 10: hidden,3: hint. They are the easiest, which comprises 10 nurses with one skill category and 3 to 4 different contract types, and the daily shifts are 4 for 28 days scheduling period.

In addition, the medium track contains 18 problem instances, which are grouped into 5: early, 5: late, 5: hidden, 3: hint. They are more complicated than the sprint track problem instances, which includes 30-31 nurses with 1 or 2 skills and 4 or 5 different contracts. The daily shifts are 4 or 5 shifts over 28 days scheduling period. Lastly, the long track includes 18 datasets, which are classified into 5: early, 5: late, 5: hidden, 3: hint. They are being referred to as hardest, which contains 49-50 nurses with 2 skills and 3 or 4 different contracts. The daily shifts are 5 shifts for 28 days scheduling period.

The parameters settings of the proposed adapted ABC are selected based on our preliminary experiments over NRP, where solution number (SN) is fixed at 10 while limit and MCN are set 100 and 10000 respectively. Table II shows the experimental results produced by the proposed technique for 69 problem instances of the INRC2010 datasets. The numbers in Table II refer to the penalty cost for the violations of the soft constraints (lowest is the best), which is computed based on the objective cost as shown in Eq. (4). Similarly, as shown in Table II, the best results obtained by the adapted ABC on 69 instances of INRC2010 are compared with those achieved by the six state-of-the-arts methods, which are listed as follows:

- T_1 Global Best Harmony Search Algorithm (Awadallah et al. [20])
- T_2 Adaptive tabu search with restart strategy (Lu and Hao [29])
- T_3 Integer programming with set of neighbourhood structures (Valouxis et al. [26])
- T_4 Variable Depth Search Algorithm and Branch and Price Algorithm (Burke and Curtois [27])
- T_5 Hyper-heuristic combined with a greedy shuffle approach (Bilgin et al. [30])
- T_6 Constraint Optimization Solver (Nonobe [28])

Basically, the proposed adapted ABC produced are very competitive results in comparison with those achieved by the six existing methods in all 69 problem instances of INRC2010 dataset. This is initial research of adapting ABC algorithm to the INCR2010 dataset. However, it is observed during run of experiment that the proposed methods suffers stagnation in local optima as well as encountering premature convergence. These shortcomings shall be addressed in our next research.

V. Conclusion

In this paper, an adaption of Artificial Bee Colony Algorithm is presented for tackling the NRP. As the results have shown in Table II, the adapted algorithm is capable of solving nurse rostering problem. Although the results produced by the algorithm in this study are compared with

TABLE II
EXPERIMENTAL RESULTS OF ADAPTED ABC AND SOME COMPARATIVE TECHNIQUES THAT WORKED ON INRC2010 DATASET

	Adapted ABC			T_1	T_2	T_3	T_4	T_5	T_6
	Best	**Mean**	**Std.**	**Best**	**Best**	**Best**	**Best**	**Best**	**Best**
Sprint_early01	62	63.9	1.9	58	56	56	56	57	56
Sprint_early02	64	65.3	1.1	60	58	58	58	59	58
Sprint_early03	58	61.6	3.9	53	51	51	51	51	51
Sprint_early04	66	67.9	1.8	62	59	59	59	60	59
Sprint_early05	63	63.6	0.5	59	58	58	58	58	58
Sprint_early06	58	59.7	2.0	56	54	54	54	54	54
Sprint_early07	61	62	0.8	58	56	56	56	56	56
Sprint_early08	58	62	3.7	57	56	56	56	56	56
Sprint_early09	58	60.4	2.2	57	55	55	55	55	55
Sprint_early10	57	59.7	2.2	53	52	52	52	52	52
Sprint_hidden01	44	46.2	1.3	41	32	33	-	-	-
Sprint_hidden02	38	42.8	5.0	35	32	32	-	-	-
Sprint_hidden03	71	74	3.1	70	62	62	-	-	-
Sprint_hidden04	76	77.6	1.8	79	66	67	-	-	-
Sprint_hidden05	65	68.4	3.0	62	59	59	-	-	-
Sprint_hidden06	161	182.6	15.8	202	130	134	-	-	-
Sprint_hidden07	178	193	18.1	196	153	153	-	-	-
Sprint_hidden08	245	252.3	7.5	266	204	209	-	-	-
Sprint_hidden09	371	379.6	9.3	273	338	338	-	-	-
Sprint_hidden10	327	344.7	19.1	346	306	306	-	-	-
Sprint_late01	49	51.9	3.2	45	37	37	37	40	37
Sprint_late02	52	53.5	1.6	49	42	42	42	44	42
Sprint_late03	56	58.5	3.0	55	48	48	48	50	48
Sprint_late04	89	100.5	6.6	104	73	75	75	81	76
Sprint_late05	53	53.8	1.0	51	44	44	44	45	45
Sprint_late06	47	48.1	1.5	43	42	42	42	42	42
Sprint_late07	52	57.8	5.5	60	42	42	42	46	43
Sprint_late08	17	23.8	8.3	17	17	17	17	17	17
Sprint_late09	17	26.2	5.4	17	17	17	17	17	17
Sprint_late10	56	57.5	1.6	54	43	43	43	46	44
Sprint_hint01	85	93.5	6.6	101	-	-	-	78	-
Sprint_hint02	57	59.9	2.6	59	-	-	-	47	-
Sprint_hint03	74	77.6	7.1	77	-	-	-	57	-
Medium_early01	260	266.9	5.3	270	240	240	244	242	241
Medium_early02	261	267	4.8	275	240	240	241	241	240
Medium_early03	259	267.4	6.2	265	236	236	238	238	236
Medium_early04	257	264.7	8.6	263	237	237	240	238	238
Medium_early05	329	333.6	6.5	334	303	303	308	304	304
Medium_hidden01	188	200.7	7.7	253	117	130	-	-	-
Medium_hidden02	284	298.1	10.9	361	220	221	-	-	-
Medium_hidden03	64	68	3.3	93	35	36	-	-	-
Medium_hidden04	100	105.4	4.1	135	79	81	-	-	-
Medium_hidden05	201	211.1	11.6	275	119	122	-	-	-
Medium_late01	206	223.4	11.6	254	164	158	187	163	176
Medium_late02	52	54.3	2.9	72	20	18	22	21	19
Medium_late03	70	72.6	2.1	75	30	29	46	32	30
Medium_late04	65	70.8	4.3	79	36	35	49	38	37
Medium_late05	178	192	11.9	238	117	107	161	122	125
Medium_hint01	69	77.2	7.9	89	-	-	-	40	-
Medium_hint02	141	151.3	10.8	194	-	-	-	91	-
Medium_hint03	187	225.6	42.5	242	-	-	-	144	-
Long_early01	242	247.2	3.8	256	197	197	198	197	197
Long_early02	277	284.1	5.2	299	222	219	223	220	219
Long_early03	269	277.4	5.0	286	240	240	242	240	240
Long_early04	337	346.6	8.3	356	303	303	305	303	303
Long_early05	327	332.1	5.3	337	284	284	286	284	284
Long_hidden01	445	457	11.2	747	346	363	-	-	-

Long_hidden02	130	132.4	2.1	225	89	90	-	-	-
Long_hidden03	59	62.4	3.5	121	38	38	-	-	-
Long_hidden04	47	50.9	2.8	134	22	22	-	-	-
Long_hidden05	76	81.5	4.0	146	45	41	-	-	-
Long_late01	288	296.9	6.5	601	237	235	286	241	235
Long_late02	293	311.5	30.7	596	229	229	290	245	229
Long_late03	306	311.1	5.8	585	222	220	290	233	220
Long_late04	303	313.7	9.6	621	227	221	280	246	221
Long_late05	141	151.5	9.6	393	83	83	110	87	83
Long_hint01	52	56.1	2.4	134	-	-	-	33	-
Long_hint02	39	44.7	8.8	102	-	-	-	17	-
Long_hint03	116	122.6	7.4	375	-	-	-	55	-

those produced by other state-of-the-arts techniques, which show the techniques is very competitive. In addition, the proposed algorithm produced comparable results on NRP, further improvement could still be made in the area of exploitation in order to enhance its performance while tackling the NRP. Therefore, our future work will focus on the enhancement of this technique in the following areas:

- To improve the adapted ABC by introducing more powerful and more structure local search mechanisms to handle specific soft constraints violations while tackling the NRP.
- To integrate adapted ABC algorithm with components of other metaheuristic algorithms.

REFERENCES

[1] John J Bartholdi III, “A guaranteed-accuracy round-off algorithm for cyclic scheduling and set covering”, Operations Research, vol. 29, no. 3, pp. 501–510, 1981.

[2] Harvey H Millar and Mona Kiragu, “Cyclic and non-cyclic scheduling of 12 h shift nurses by network programming”, European journal of operational research, vol. 104, no. 3, pp. 582–592, 1998.

[3] Anthony Wren, “Scheduling, timetabling and rosteringa special relationship? The Practice and Theory of Automated Timetabling I: Selected Papers from 1st International Conference on the Practice and Theory of Automated Timetabling (PATAT I), Edinburgh, UK, 1996, pp 46-75.

[4] Andrew J Mason and Mark C Smith, “A nested column generator for solving rostering problems with integer programming”, in International conference on optimisation: techniques and applications, Curtin University of Technology Perth, Australia, 1998, pp. 827-834.

[5] Broos Maenhout and Mario Vanhoucke, “Branching strategies in a branch-and-price approach for a multiple objective nurse scheduling problem”, Journal of Scheduling, vol. 13, no. 1, pp. 77–93, 2010.

[6] M Naceur Azaiez and SS Al Sharif, “A 0-1 goal programming model for nurse scheduling”, Computers & Operations Research, vol. 32, no. 3, pp. 491–507, 2005.

[7] Gareth Beddoe and Sanja Petrovic, “Enhancing case-based reasoning for personnel rostering with selected tabu search concepts”, Journal of the Operational Research Society, vol. 58, no. 12, pp. 1586–1598, 2007.

[8] Gareth Beddoe, Sanja Petrovic, and Jingpeng Li, “A hybrid metaheuristic case-based reasoning system for nurse rostering”, Journal of Scheduling, vol. 12, no. 2, pp. 99–119, 2009.

[9] Rong Qu and Fang He, “A hybrid constraint programming approach for nurse rostering problems”, in Applications and innovations in intelligent systems XVI, Springer, London, 2009 pp. 211–224..

[10] Haibing Li, Andrew Lim, and Brian Rodrigues, “A hybrid AI approach for nurse rostering problem”, in Proceedings of the ACM symposium on Applied computing (SAC 2003), Melbourne, Florida, 2003, pp. 730–735.

[11] Kathryn A Dowsland, “Nurse scheduling with tabu search and strategic oscillation”, European journal of operational research, vol. 106, no. 2, pp. 393–407, 1998.

[12] Edmund Burke, Patrick De Causmaecker, and Greet Vanden Berghe, “A hybrid tabu search algorithm for the nurse rostering problem”, in Simulated evolution and learning, Springer, Canberra, Australia, 1999, pp. 187–194.

[13] RN Bailey, KM Garner, and MF Hobbs, “Using simulated annealing and genetic algorithms to solve staff scheduling problems”, Asia-Pacific Journal of Operational Research, vol. 14, no. 2, pp. 27–43, 1997.

[14] Edmund K Burke, Timothy Curtois, Gerhard Post, Rong Qu, and Bart Veltman, “A hybrid heuristic ordering and variable neighbourhood search for the nurse rostering problem”, European Journal of Operational Research, vol. 188, no. 2, pp. 330–341, 2008.

[15] Edmund Burke, Patrick De Causmaecker, Sanja Petrovic, and Greet Vanden Berghe, “Variable neighborhood search for nurse rostering problems”, in Metaheuristics: computer decision-making, pp. 153–172. Springer, 2004.

[16] Walter J Gutjahr and Marion S Rauner, “An aco algorithm for a dynamic regional nurse-scheduling problem in austria”, Computers & Operations Research, vol. 34, no. 3, pp. 642–666, 2007.

[17] Uwe Aickelin and Kathryn A Dowsland, “An indirect genetic algorithm for a nurse-scheduling problem”, Computers & Operations Research, vol. 31, no. 5, pp. 761–778, 2004.

[18] Margarida Moz and Margarida Vaz Pato, “A genetic algorithm approach to a nurse rerostering problem”, Computers & Operations Research, vol. 34, no. 3, pp. 667–691, 2007.

[19] Mohammed A Awadallah, Ahamad Tajudin Khader, Mohammed Azmi Al-Betar, and Asaju La'aro Bolaji, “Nurse scheduling using modified harmony search algorithm”, in in sixth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2011), IEEE, Penang, Malaysia, 2011, pp. 58–63.

[20] Mohammed A Awadallah, Ahamad Tajudin Khader, Mohammed Azmi Al-Betar, and Asaju La'aro Bolaji, “Global best harmony search with a new pitch adjustment designed for nurse rostering”, Journal of King Saud University-Computer and Information Sciences, vol. 25, no. 2, pp.145-162, 2013.

[21] Tai-Hsi Wu, Jinn-Yi Yeh, and Yueh-Min Lee, “A particle swarm optimization approach with refinement procedure for nurse rostering problem”, Computers & Operations Research, vol. 52, pp.52-63, 2014.

[22] Burak Bilgin, Patrick De Causmaecker, and Greet Vanden Berghe, “A hyperheuristic approach to belgian nurse rostering problems”, in Proceedings of the 4th Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA 2009), Dublin, 2009, pp. 693–695.

[23] Brenda Cheang, Haibing Li, Andrew Lim, and Brian Rodrigues, “Nurse rostering problems - a bibliographic survey”, European Journal of Operational Research, vol. 151, no. 3, pp. 447–460, 2003.

[24] Edmund K Burke, Patrick De Causmaecker, Greet Vanden Berghe, and Hendrik Van Landeghem, “The state of the art of nurse rostering”, Journal of scheduling, vol. 7, no. 6, pp. 441–499, 2004.

[25] Jorne Van den Bergh, Jeroen Beli¨en, Philippe De Bruecker, Erik Demeulemeester, and Liesje De Boeck, “Personnel scheduling: A literature review”, European Journal of Operational Research, vol. 226, no. 3, pp. 367–385, 2013.

[26] Christos Valouxis, Christos Gogos, George Goulas, Panayiotis Alefragis, and Efthymios Housos, “A systematic two phase approach for the nurse rostering problem”, European Journal of Operational Research, vol. 219, no. 2, pp. 425–433, 2012.

[27] Edmund K Burke and Tim Curtois, “An ejection chain method and a branch and price algorithm applied to the instances of the first international nurse rostering competition, 2010”, in Proceedings of the 8th International Conference on the Practice and Theory of Automated Timetabling PATAT. Citeseer, 2010, vol. 10, pp. 13 – 23.

[28] Koji Nonobe, “Inrc2010: An approach using a general constraint optimization solver”, Technical Report, INRC2010 (http://kuleuventortrijk.be/nrcompetition).

[29] Zhipeng Lu and Jin-Kao Hao, "Adaptive neighborhood search for nurse rostering", European Journal of Operational Research, vol. 218, no. 3, pp. 865-876, 2012.

[30] Burak Bilgin, Patrick De Causmaecker, Benoˆıt Rossie, and Greet Vanden Berghe, "Local search neighbourhoods for dealing with a novel nurse rostering model", Annals of Operations Research, vol. 194, no. 1, pp. 33-57, 2012.

[31] Ademir Aparecido Constantino, Dario Landa-Silva, Everton Luiz de Melo, Candido Ferreira Xavier de Mendonc¸a, Douglas Baroni Rizzato, and Wesley Rom˜ao, "A heuristic algorithm based on multi-assignment procedures for nurse scheduling", Annals of Operations Research, vol. 218, no. 1 pp. 165–183, 2014.

[32] Mohammed A Awadallah, Ahamad Tajudin Khader, Mohammed Azmi Al-Betar, and Asaju La'aro Bolaji, "Nurse rostering using modified harmony search algorithm", in Swarm, Evolutionary, and Memetic Computing, Springer, India, 2011, pp. 27–37.

[33] Mohammed A Awadallah, Ahamad Tajudin Khader, Mohammed Azmi Al-Betar, and Asaju La'aro Bolaji, "Harmony search with greedy shuffle for nurse rostering", International Journal of Natural Computing Research (IJNCR), vol. 3, no. 2, pp. 22–42, 2012.

[34] Mohammed A Awadallah, Ahamad Tajudin Khader, Mohammed Azmi Al-Betar, and Asaju La'aro Bolaji, "Hybrid harmony search for nurse rostering problems", in Computational Intelligence in Scheduling (SCIS), 2013 IEEE Symposium, Singapore, 2013, pp. 60–67.

[35] Mohammed A Awadallah, Ahamad Tajudin Khader, Mohammed Azmi Al-Betar, and Asaju La'aro Bolaji, "Harmony search with novel selection methods in memory consideration for nurse rostering problem", Asia-Pacific Journal of Operational Research, vol. 31, no. 3, pp. 1–39, 2013.

[36] D. Karaboga, "An idea based on honey bee swarm for numerical optimization", Techn. Rep. TR06, Erciyes Univ. Press, Erciyes, 2005.

[37] D. Teodorovi´c and M. DellOrco, "Bee colony optimization - a cooperative learning approach to complex transportation problems", in Advanced OR and AI Methods in Transportation. Proceedings of the 10th Meeting of the EURO Working Group on Transportation, Poznan, Poland. Citeseer, 2005, pp. 51–60.

[38] Asaju La'aro Bolaji, A.T. Khader, M.A. Al-betar, and M. Awadallah, "The effect of neighborhood structures on examination timetabling with artificial bee colony", in Practice and Theory of Automated Timetabling IX, Son, Norway, 2012, pp. 131–144.

[39] Asaju La'aro Bolaji, Ahamad Tajudin Khader, Mohammed Azmi Al-Betar, and Mohammed A Awadallah, "University course timetabling using hybridized artificial bee colony with hill climbing optimizer", Journal of Computational Science, vol. 5, no. 5, pp. 809–818, 2014.

[40] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: artificial bee colony (abc) algorithm and applications", Artificial Intelligence Review, vol. 42, no. 1, pp. 1–37, 2012.

[41] A.L. Bolaji, A.T. Khader, M.A. Al-Betar, and M.A. Awadallah, "Artificial bee colony, its variants and applications: a survey", Journal of Theoretical & Applied Information Technology (JATIT), vol. 47, no. 2, pp. 434–459, 2013..