International Journal of Electrical and Computer Engineering (IJECE)
Vol. 11, No. 1, February 2021, pp. 471~480
ISSN: 2088-8708, DOI: 10.11591/ijece.v11i1.pp471-480

A novel population-based local search for nurse rostering problem

Anmar Abuhamdah[1], Wadii Boulila[2], Ghaith M. Jaradat[3], Anas M. Quteishat[4], Mutasem K. Alsmadi[5], Ibrahim A. Almarashdeh[6]
[1]MIS Department, Faculty of Business Administration, Taibah University, Saudi Arabia
[2]IS Department, College of Computer science and Engineering, Taibah University, Saudi Arabia
[2]RIADI Laboratory, National School of Computer Science, Manouba University, Tunisia
[3]Department of Computer Science, Faculty of Computer Science and Information Technology, Jerash University, Jordan
[4]Department of Computer and Network Engineering, Faculty of Engineering Technology, Al-Balqaa Applied University, Jordan
[4]Electrical and Computer Engineering, Faculty of Engineering, Sohar University, Oman
[5,6]MIS Department, College of Applied Studies and Community Service, Imam Abdulrahman Bin Faisal University, Saudi Arabia

Article Info

Article history:

Received May 17, 2020
Revised Jun 30, 2020
Accepted Aug 5, 2020

Keywords:

Diversifications
Gravitational emulation
Local search approaches
Nurse rostering problem
Population-based approaches

ABSTRACT

Population-based approaches regularly are better than single based (local search) approaches in exploring the search space. However, the drawback of population-based approaches is in exploiting the search space. Several hybrid approaches have proven their efficiency through different domains of optimization problems by incorporating and integrating the strength of population and local search approaches. Meanwhile, hybrid methods have a drawback of increasing the parameter tuning. Recently, population-based local search was proposed for a university course-timetabling problem with fewer parameters than existing approaches, the proposed approach proves its effectiveness. The proposed approach employs two operators to intensify and diversify the search space. The first operator is applied to a single solution, while the second is applied for all solutions. This paper aims to investigate the performance of population-based local search for the nurse rostering problem. The INRC2010 database with a dataset composed of 69 instances is used to test the performance of PB-LS. A comparison was made between the performance of PB-LS and other existing approaches in the literature. Results show good performances of proposed approach compared to other approaches, where population-based local search provided best results in 55 cases over 69 instances used in experiments.

Corresponding Author:

Anmar Fakhri Abuhamdah,
MIS Department, Faculty of Business Administration,
Taibah University,
Janadah Bin Umayyah Road Tayba, Medina 42353, Saudi Arabia.
Email: aabuhamdah@taibahu.edu.sa ; anmar_81@hotmail.com

1. INTRODUCTION

Nurse rostering problem (NRP) involves assigning a set of nurses to different shifts subject to a variety of constraints until a complete roster is constructed [1, 2]. Constraints in the NRP problem can be categorized as hard and soft [3-5]. The main goal, when allocating nurses to various shifts is to fulfill the hard constrains (i.e. feasible rosters) and try to minimize the soft constraints as much as possible with

penalization (in order to produce a high-quality nurses shifts). The smaller value indicates a better quality of nurse rostering shifting. NRP is an NP-hard problem due to the difficulty of finding optimal shifts [3, 4]. However, finding a rostering with a good nurse quality relies on the approach used during the search [3-5].

Recently, several methods have been applied to solve NRP [6-18]. Jaradat *et al*. [6] proposed the hybrid elitist-ant system algorithm. The proposed algorithm aims to increase the diversity of the search space in the elitist-ant system algorithm by integrating it with external memory.

Rajeswari *et al*. [7] used a modified nelder-mead method in the bee colony optimization algorithm for multi-objective mathematical programming. The methodology is a combination of specific local search, honeybee decision making, and multi-agent particle environment system. Awadallah *et al*. [8] proposed a hybrid-harmony search algorithm with hill climbing for increasing the exploitation mechanism. The authors modified the memory to accelerate the convergence rate by using the global swarm optimization concept. Santos *et al*. [9] used integer-programming techniques of the compact and monolithic formulation. The proposed technique is distributed into two pools, the first one improves the dual bounds for rapid solution production to lower the optimal solution, while the second is used to speed-up the production near-optimal. Awadallah *et al*. [10] replaced the bee operator in the artificial bee colony algorithm by the hill-climbing optimizer to propose a hybridization that exploits the search space for good solution quality. Burke and Curtois [11], proposed new approaches for branch and price algorithm and an ejection chain, where they integrated both algorithms by using the dynamic programming method. Lü and Hao [5] proposed an adaptive neighborhood search by using two different neighborhood moves and adaptively alterations between three intensification and diversification search strategies referring to the search history. Bilgin *et al*. [12] presented a general hyper-heuristic approach that can work into two different timetabling healthcare problems (i.e. patient admission and nurse rostering) and provided a set of a low-level heuristic for each problem. Valouxis *et al*. [13] proposed a strategy based on phases to produce a good quality solution. The first phase deals with each nurse and each day of the week and the second deals with a specific allocated daily shift. Nonobe [14] used metaheuristic based on the constraint-optimization problem to find suitable values to variables that have constraint violations' weight. The author adopted the tabu search to improve the performance of controlling dynamically the tabu tenure. He used constrain weights to examine solutions during the neighborhood search.

Most of these approaches are hybridization between population-based and local-based approaches. In general, the main advantage of population-based approaches is their ability to explore the research space, whereas local-search approaches are more adapted to exploit the research space [19-35]. Therefore, combining the population-based and local-search approaches will help taking advantage of these two approaches and produces a good approach for solving NRP. However, combining these two approaches involves considering an increasing number of parameters. This constitutes a new challenge for most hybridization approaches. Hence, most researches move toward proposing other approaches with fewer parameters' tuning for NRP. The main contribution of this paper is to use population-based local search (PB-LS) [36], which is applied in another optimization problem (i.e. Course timetabling) to the nurse rostering problem. PB-LS has proved its efficiency with fewer parameters; hence, it can be applied to the NRP, which has different neighborhood structures.

Our work aims to study the performance of PB-LS over the nurse rostering problem. The proposed approach will help to obtain a good quality solution for nurse rostering shifting. In order to evaluate the efficiency of PB-LS, we compare the performance of PB-LS with other approaches applied to NRP in the literature. Sixty-nine datasets were used and results indicate that PB-LS can produce good results.

Following the introduction, the paper is structured as follows. Section 2 presents a description of the dataset used in this paper. Section 3 details the proposed methodology. Section 4 presents the results obtained by applying PB-LS to the considered dataset and discuss its findings. Finally, Section 5 concludes the paper and discusses perspectives for the proposed work.

2. DATASET DESCRIPTION

A 69 standard NRP benchmark datasets are used to test the performance of the proposed method. Datasets were obtained from PATAT 2010, international nurse rostering completion (INRC2010, https://www.kuleuven-kulak.be/nrpcompetition/instances) [37] and another version introduced in [38]. NRC2010 datasets are classified into four groups, toy; sprint; medium and long instances. Toy instances are provided for testing purposes, where the other three sprint; medium and long instances are provided for competition purpose, and used by researchers to evaluate their approaches.

Each dataset of the three competition datasets contains three groups of instances, the early, late, and hidden instance groups. Later on, an update with extra group added to them named as hint group. Table 1 summarizes the datasets that are used in this paper.

Table 1. INRC2010 datasets summarization

Instances	Early instances	Hidden instances	Hint instances	Late instances
Sprint run	10	10	3	10
Medium run	5	5	3	5
Long run	5	5	3	5

3. METHODOLOGY

In this paper, a population-based local search approach (PB-LS) is proposed [36] for nurse rostering problems (NRP). PB-LS is assembled of gravitational emulation local search (GELS), based on the natural principle of gravitational attraction [39]. PB-LS is proposed to overcome some of the population based approaches limitations such as [36]:

- Using a population of solutions in local search to increase the capability of the intensification process and to overcome the weakness of the intensification process, which leads to non-significant improvements.
- Using Memory Guide with elite solutions for useful information (in descending order) to overcome the limitation of the approaches that did not employ memory guidance for the search. In addition, this allows overcoming the dependency on randomization that requires a repair mechanism to be efficient in constrained problems and overcoming the issue of randomly updating the population.
- Less parameterized and systemically updating the population.

PB-LS proves its efficiency over an NP-hard optimization problem of university course timetabling problem [36]. This motivated us to use this method for NRP by using different neighborhood structures. PB-LS starts with an initial population and tries to minimize soft constraints iteratively by exploring their neighbor solutions. These solutions are obtained by using one or more neighborhood structures over the current solution. For more details about the constraints Hint, Hard and Soft presented in the INRC2010 datasets and their mathematical formulations, readers can refer to [5]. As in [6], four groups of neighborhood structures are used:

- Single shift per day neighborhood structure associated with hard constraint.
- Weekend, personal request, alternative qualifications, overtime, and under time are the most violated constraints associated with soft constraints.
- Shuffle, greedy shuffle, core shuffle to swap large sections of personal schedules.
- Shake a shift, weekend, and two people for solution shaking.

PB-LS calculates the gravitational force value (F, assuming minimization problem) by calculating the difference between two objects; the trial solutions objective function values (i.e. Ts) and the current solution objective function value (i.e. Cs) as presented in (1).

$$F = Cs - Ts \quad (1)$$

Figure 1 shows the pseudo-code for the PB-LS approach for NRP. Below the description of some terms that are used in Figure 1.

- Initial solution: *Si*; and the quality of Si denoted by *f(Si)*
- Best obtained solution: Sb and the quality of Si denoted by f(Sb)
- The maximum number of iterations denoted by N.max
- Number of iterations to reset the solution directions to update denoted by *R.iter*
- Force value for gravity denoted by *force*
- N^{th} velocity vector memory to provide direction value denoted by VV_n
- Number of shaking neighborhood denoted by *Nsn*

The PB-LS algorithm starts with a zero direction value (i.e. zero velocity) that initializes the search, and then this value is updated throughout the search process. For better shifting, the search space is administered by the force value (using (1)), and then the search space is intensified using any local search algorithm. PB-LS uses the MPCA-ARDA algorithm [36] as a local search due to its ability of complementary exploration and exploitation. MPCA-ARDA is proposed in [40] as hybridization between multi-neighborhood particle collision algorithm and adaptive randomized descent algorithm.

In step 1 (i.e. Initialization phase), we initialize all the required parameters mentioned in Table 2. Then, the initial solution for the memory of the velocity vector *vv* is generated by applying shaking neighborhoods. Initially, solutions (vv_1,…, vv_N) are produced based on the number of neighborhoods and their directions are set to zero in the *vv* memory.

```
Step 1: Initialization Phase
  Given an initial candidate solution Si with f(Si) as the quality of Si;
  Set Sb = Si , f(Sb)= f(Si), Nsn = number of shaking neighbourhood;
  Generate Nsn neighbours of Sb (one neighbour from each shaking neighbourhood);
  Assign each generated neighbour to the velocity vector, vv;
  Set all the directions and un-improve counters (UnImprove) for each vector in vv = 0;
  Set N.max; //stopping condition (e.g. see Table 2);
  Set R.iter; // the number of iterations to reset the direction & update solutions;
 Initialize the required parameters for the Local Search Method; //depend on what local
  search is used
Step 2: Improvement (Iterative) Phase
Iterations =1;
repeat
   Arrange the vector in vv in descending order based on their direction value;
   2.1 If the direction is clear // No duplication in the direction values
         Set Si = vv1; // Select the best direction
         Set n=1;
    a) Apply Local Search on So to produce Si*; //any local search can be applied
         Calculate the force value for the vvn using equation 1; //positive or negative force
         with Si as a current solution and Si*as a candidate solution.
     Update the direction value by adding force value to the direction of the vvn;
     If f(Si*) is better than f(Sb), then Sb= Si*;
     If f(Si*) is better than f(Si), then set vvn = Si*;
     Else
        Increase the UnImprovek by one;
        If UnImprovek == R.iter;
           Set the direction of vvn = 0;
           UnImproven=0;
           Randomly generate Nsn neighbours of Sb (one neighbour from each shaking
           neighbourhood) and set vvn = Sb* (the best neighbour of Sb);
        End If
     End Else;
   End if

   2.2 Else ; // Improve all solutions that has similar direction value
      While (direction not clear) //some vv have similar direction value
         Select the vv* (the vv that have similar direction values with other vv), set
         n=index of vv* , set Si = vv* and apply Step 2.1a);
     End while
   End Else;
  iterations=iterations+1;
until iterations> N.iters (termination condition is met)
Step 3: Termination phase (N.max termination condition is met)
     Return the best found solution Sb
```

Figure 1. Pseudocode for PB-LS approach [36] over nurse rostering problem

Table 2. Parameters used in PB-LS for NRP

Parameter	Value
N.max	Termination Criteria (maximum iterations) equal to 100,000
R.iter	Iterations number to reset or to retune the directions equal to10
Nsn	Shaking neighborhood structures
Local Search	MPCA-ARDA (iterations number equal to10)

In step 2 (i.e. Improvement phase), *vv* memory is reordered in decreasing order based on their direction values. The solutions are compared based on their direction values, and the solution with the highest direction value is selected for further improvement prior to other solutions in *vv*.

a. In step 2.1, if all direction values are different, we select the highest solution direction value and set $S_i=vv_1$, then MPCA-ARDA is applied to generate S_i^* until the stopping criteria are met. MPCA-ARDA uses different shaking neighborhood. Later, the force value is calculated using (1) (i.e. Cs = S_i and Ts=S_i^*) and is added (negative or positive) to the stored direction value of vv_1. The best solution (i.e. *Sb*) is updated with S_i^* in case of $f(S_i^*)$ (i.e. the quality of S_i^*) is better than *f(Sb)* (i.e. the quality of Sb). If the quality of S_i^* is better than S_i, we replace vv_1 with S_i^*. Otherwise, the unimproved counter is increased ($UnImprove_1$) by one for the selected solution. In case *UnImprove1* is equal to pre-set unimproved iterations (i.e. *R.iter*), the direction of vv_n is returned to zero and vv_1 solution is changed with the best neighbor solution generated from *Sb* random neighbors (i.e. Sb*) by shaking the neighborhoods. This practicability tries to escape from local optima and attempts to expand the search space.
b. In step 2.2, if the direction values are similar for all *vv* solutions, step 2.1.a is executed to distinguish the solutions of similar direction values. This tries to preserve a set of diverse solutions.

4. RESULTS

The proposed approach is experimented for 25 runs (as suggested in [6]) through 69 instances that were announced in INRC2010 (https://www.kuleuven-kulak.be/nrpcompetition/instances) for 100,000

iterations. The machine used is an Intel Core i5 3.2 GHz processor, 8 GB RAM, and the code implemented using java language over NetBeans IDE v 8.2. As stated in INRC2010, the instances require a solution within approximately 10 seconds for the sprint instances, 10 minutes for the medium instances, and 10 hours for the long instances.

However, different factors affect the performance of the machine such as memory, clock speed and the operating system. Therefore, the results of the runs are obtained at different times. As depicted in [5], simulations in our approach are performed under relaxing timeout conditions, and we employ 1000 seconds for the sprint instances, 5000 seconds for the medium instances and 20 hours for the long instances. As shown in Table 2, PB-LS employs four parameters:

a. The termination criteria (*N.max*), which is presented in [6] and takes into consideration the relax timeout condition.
b. Iteration number to reset (or retune) the directions (*R.iter*), which is presented in [2].
c. Shaking- neighborhood structures (*Nsn*), which is presented in [6];
d. Iteration number for the local search (in this case, MPCA-ARDA used as a local search with 10 iterations), which is presented in [36].

In order to evaluate the performance of PB-LS, a comparison is made between PB-LS performance and results of similar methods based on their published works. Table 3 shows a comparison based on sprint instances, Table 4 shows a comparison based on medium instances and Table 5 shows a comparison based on long instances.

In Tables 3-5, the best result obtained by PB-LS (denoted as *Best*) and the rank of PB-LS (dented as *Rank*) are illustrated and compared to the other approaches. The best results so far are depicted by bold color. Meanwhile, in Tables 3-5, the empty cells with a symbol (-) denote that the algorithm is not applied to the corresponding instance, for example, R4, R6, R7, R9 and R10 did not apply their algorithms to the 9 hint instances, and R6, R8 and R10 did not apply their algorithms to the 20 hidden instances.

Table 3. Comparison between PB-LS and similar methods in the literature using sprint instances

Dataset	PB-LS		R1	R2	R3	R4	R5	R6	R7	R8	R9	R10
	Rank	Best										
sprint_early_01	8	57	57	56	58	56	56	56	56	57	56	56
sprint_early_02	Same	58	59	59	64	58	58	58	58	59	58	58
sprint_early_03	Same	51	51	51	59	51	51	51	51	51	51	51
sprint_early_04	Same	59	59	59	67	59	59	59	59	60	59	59
sprint_early_05	Same	58	58	58	63	58	58	58	58	58	58	58
sprint_early_06	2	54	54	53	58	54	54	54	54	54	54	54
sprint_early_07	Same	56	56	56	61	56	56	56	56	56	56	56
sprint_early_08	Same	56	56	56	58	56	56	56	56	56	56	56
sprint_early_09	Same	55	55	55	61	55	55	55	55	55	55	55
sprint_early_10	Same	52	52	52	58	52	52	52	52	52	52	52
sprint_hidden_01	Same	32	32	32	46	32	32	-	32	-	33	-
sprint_hidden_02	Same	32	32	32	44	32	32	-	32	-	32	-
sprint_hidden_03	Same	62	62	62	78	62	62	-	62	-	62	-
sprint_hidden_04	Same	66	66	66	78	66	66	-	66	-	67	-
sprint_hidden_05	Same	59	59	59	69	59	59	-	59	-	59	-
sprint_hidden_06	Same	130	130	134	169	130	130	-	130	-	134	-
sprint_hidden_07	Same	153	153	153	187	153	153	-	153	-	153	-
sprint_hidden_08	Same	204	204	204	240	204	204	-	204	-	209	-
sprint_hidden_09	Same	338	338	338	372	338	338	-	338	-	338	-
sprint_hidden_10	Same	306	306	306	322	306	306	-	306	-	306	-
sprint_hint_01	2	75	78	73	90	-	75	-	-	78	-	-
sprint_hint_02	2	46	47	43	56	-	46	-	-	47	-	-
sprint_hint_03	2	50	50	49	69	-	50	-	-	57	-	-
sprint_late_01	Same	37	37	37	52	37	37	37	37	40	37	37
sprint_late_02	2	42	42	41	56	42	42	42	42	44	42	42
sprint_late_03	2	48	48	45	60	48	48	48	48	50	48	48
sprint_late_04	2	73	73	71	95	73	73	75	73	81	75	76
sprint_late_05	Same	44	44	46	57	44	44	44	44	45	44	45
sprint_late_06	Same	42	42	42	52	42	42	42	42	42	42	42
sprint_late_07	Same	42	42	44	55	42	44	42	42	46	42	43
sprint_late_08	Same	17	17	17	19	17	17	17	17	17	17	17
sprint_late_09	Same	17	17	17	17	17	17	17	17	17	17	17
sprint_late_10	Same	43	43	43	54	43	43	43	43	46	43	44

Note: R1: Hybrid Elitist-Ant System [6], R2: Directed Bee Colony Optimization Algorithm [7], R3: Hybridization of harmony search with hill-climbing [8], R4: Integer programming techniques [9], R5: A hybrid artificial bee colony [10], R6: New approach for branch and price algorithm and an ejection chain [11], R7: Adaptive neighborhood search [5], R8: Hyper-heuristic approach [12], R9: A systematic two-phase approach [13] and R10: A general constraint optimization [14].

Table 4. Comparison between PB-LS and similar methods in the literature using medium instances

Dataset	PB-LS		R1	R2	R3	R4	R5	R6	R7	R8	R9	R10
	Rank	Best										
medium_early_01	Same	240	241	245	280	240	245	244	240	242	240	241
medium_early_02	Same	240	241	243	281	240	245	241	240	241	240	240
medium_early_03	Same	236	236	239	287	236	242	238	236	238	236	236
medium_early_04	Same	237	237	245	278	237	240	240	237	238	237	239
medium_early_05	Same	303	303	310	330	303	308	308	303	304	303	304
medium_hidden_01	Same	111	111	143	410	111	155	-	117	-	130	-
medium_hidden_02	Same	220	220	230	412	221	254	-	220	-	221	-
medium_hidden_03	Same	34	34	53	182	34	54	-	35	-	36	-
medium_hidden_04	Same	78	78	85	168	78	94	-	79	-	81	-
medium_hidden_05	Same	119	119	182	520	119	177	-	119	-	122	-
medium_hint_01	2	42	42	42	64	-	48	-	-	40	-	-
medium_hint_02	Same	91	91	91	133	-	94	-	-	91	-	-
medium_hint_03	2	140	140	135	187	-	140	-	-	144	-	-
medium_late_01	2	161	161	176	234	157	174	187	164	163	158	176
medium_late_02	Same	18	18	30	49	18	31	22	20	21	18	19
medium_late_03	Same	29	29	35	59	29	38	46	30	32	29	30
medium_late_04	Same	35	35	42	71	35	48	49	36	38	35	37
medium_late_05	Same	107	107	129	272	107	137	161	117	122	107	125

Table 5. Comparison between PB-LS and similar methods in the literature using long instances

Dataset	PB-LS		R1	R2	R3	R4	R5	R6	R7	R8	R9	R10
	Rank	Best										
long_early_01	2	197	198	194	339	197	197	198	197	197	197	197
long_early_02	Same	219	220	228	399	219	229	223	222	220	219	219
long_early_03	Same	240	240	240	349	240	240	242	240	240	240	240
long_early_04	Same	303	303	303	411	303	303	305	303	303	303	303
long_early_05	Same	284	284	284	383	284	284	286	284	284	284	284
long_hidden_01	Same	346	346	389	4466	346	400	-	346	-	363	-
long_hidden_02	Same	89	89	108	1071	89	117	-	89	-	90	-
long_hidden_03	Same	38	38	48	163	38	51	-	38	-	38	-
long_hidden_04	Same	22	22	27	113	22	29	-	22	-	22	-
long_hidden_05	Same	41	41	55	139	41	56	-	45	-	41	-
long_hint_01	2	40	40	40	126	-	42	-	-	33	-	-
long_hint_02	2	28	28	29	122	-	30	-	-	17	-	-
long_hint_03	Same	55	55	79	278	-	83	-	-	55	-	-
long_late_01	Same	235	235	249	588	235	257	286	237	241	235	235
long_late_02	Same	229	229	261	577	229	263	290	229	245	229	229
long_late_03	Same	220	220	259	567	220	262	290	222	233	220	220
long_late_04	Same	221	221	257	604	222	261	280	227	246	221	221
long_late_05	Same	83	83	92	329	83	102	110	83	87	83	83

As shown in Tables 3-5, results indicate that PB-LS produces very good quality solutions. The proposed approach reaches 55 optimal solutions out of 69 instances. While in other instances PB-LS obtained the second rank except in one instance (i.e. sprint_early_01).

Despite that PB-LS has obtained second place in some instances, it outperformed the same approaches in other different instances. For example, PB-LS obtained the eighth rank in sprint_early_01 and outperformed R2, R4, R5, R6, R7, R9 and R10. In addition, PB-LS outperformed R2 in sprint_early_02, R4 in medium_hidden_02, R5 in sprint_late_07, R6 in medium_early_01, R7 in medium_hidden_01, R9 in sprint_hidden_01, R10 in sprint_late_05. Moreover, PB-LS outperformed R1 in sprint_early_02, R3 in sprint_early_02, and R8 in sprint_early_02. Results indicate that PB-LS outperformed all approaches in some instances and obtained similar performances to other approaches in some cases. This confirms that PB-LS can be considered as a good approach for NRP. To better evaluate the performance of the proposed approach, Figure 2 depicts a comparison between PB-LS and other approaches refereeing to the number of best results (optimal solutions) obtained over the 69 instances.

Figure 2 shows the number of best results obtained over 69 datasets for the proposed approach and the considered existing works. The proposed approach obtained 55 best optimal solutions over 69 instances. R4 comes in the second rank with 52 best optimal solutions. R3 comes in the last rank with only one optimal solution.

Figure 3 depicts further analysis by comparing similar, worse and better results of the proposed approach over the 69 instances to the other considered approaches. In the legend, the blue color denotes that the proposed approach has the same number of better results than the existing approach. The orange color

represents the number of times that the proposed approach has the best results. The gray color represents the number of times that the existing approach has the best results. For example, the proposed approach and R1 obtained 62 similar number of best solutions. However, the proposed approach outperformed R1 in 7 cases. In addition, in all cases, PB-LS obtained several better results greater than the other existing approaches (orange bar against gray bar). Table 6 summarizes experimental results achieved by PB-LS over sprint instances, Table 7 for the medium instances, and Table 8 for the long instances.

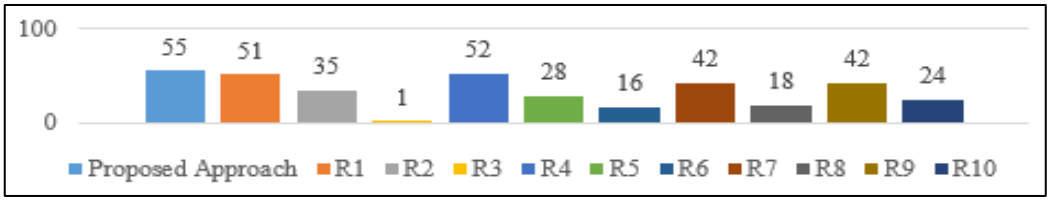

Figure 2. Comparison of the number of best results obtained over the 69 instances

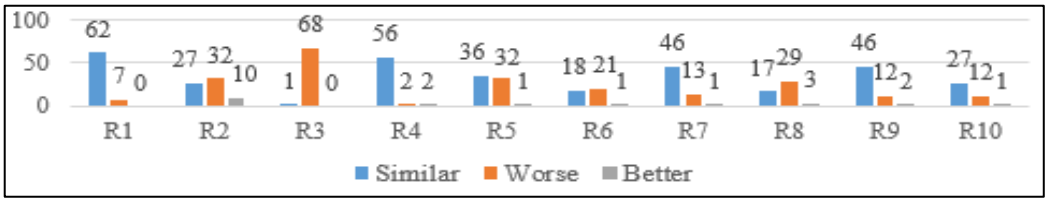

Figure 3. The number of similar, worse and better results compared to the proposed approach

Table 6. Experimental results for our approach on INRC2010 over sprint instances

Dataset	Best	avg	σ	Time
sprint_early_01	57	58.7	1.8	4*
sprint_early_02	58	58.6	1.0	3*
sprint_early_03	51	51.8	0.7	7*
sprint_early_04	59	60.1	0.8	4*
sprint_early_05	58	58.2	0.4	6*
sprint_early_06	54	54.2	0.4	3*
sprint_early_07	56	56.6	0.5	3*
sprint_early_08	56	56.4	0.4	4*
sprint_early_09	55	55.5	0.7	4*
sprint_early_10	52	52.5	0.4	6*
sprint_hidden_01	32	33.9	1.2	68
sprint_hidden_02	32	33.5	1.1	86
sprint_hidden_03	62	63.5	1.9	6*
sprint_hidden_04	66	67.2	0.7	24
sprint_hidden_05	59	59.6	0.6	8*
sprint_hidden_06	130	133.4	2.4	48
sprint_hidden_07	153	156.1	3.7	5*
sprint_hidden_08	204	205.8	1.6	71
sprint_hidden_09	338	340.2	2.8	34
sprint_hidden_10	306	306.6	1.9	7*
sprint_hint_01	75	77.2	5.4	44
sprint_hint_02	46	49.2	2.9	26
sprint_hint_03	50	54.7	6.7	35
sprint_late_01	37	38.3	1.1	23
sprint_late_02	42	43.7	1.5	18
sprint_late_03	48	50.2	0.9	8*
sprint_late_04	73	76.6	2.4	28
sprint_late_05	44	45.1	0.8	6*
sprint_late_06	42	42.5	0.6	9*
sprint_late_07	42	43.9	0.9	8*
sprint_late_08	17	17	0	5*
sprint_late_09	17	17	0	4*
sprint_late_10	43	45.2	1.2	10*

Table 7. Experimental results for our approach on INRC2010 over medium instances

Dataset	Best	avg	σ	Time
medium_early_01	240	241.6	1.7	957
medium_early_02	240	241.5	1.1	567*
medium_early_03	236	238.6	1.2	921
medium_early_04	237	238.7	1.4	249*
medium_early_05	303	305.3	1.8	353*
medium_hidden_01	111	120.8	3.8	2153
medium_hidden_02	220	233.7	8.2	4370
medium_hidden_03	34	35.9	1.9	705
medium_hidden_04	78	82.4	2.5	2081
medium_hidden_05	119	129.3	5.4	4429
medium_hint_01	42	51.8	4.6	1505
medium_hint_02	91	105.3	7.6	2453
medium_hint_03	140	151.1	8.7	2691
medium_late_01	161	172.6	6.3	3553
medium_late_02	18	21.8	2.3	2417
medium_late_03	29	32.9	2.2	583*
medium_late_04	35	36.8	0.7	1108
medium_late_05	107	116.7	6.8	4230

Table 8. Experimental results for our approach on INRC2010 over long instances

Dataset	Best	avg	σ	Time
long_early_01	197	201.4	3.7	11114*
long_early_02	219	231.3	6.3	12021*
long_early_03	240	240	0	3041*
long_early_04	303	304.9	0.7	5112*
long_early_05	284	285.6	1.2	8392*
long_hidden_01	346	355.9	5.4	21297*
long_hidden_02	89	91.6	2.8	18844*
long_hidden_03	38	40.1	1.2	14947*
long_hidden_04	22	24.9	1.5	24663*
long_hidden_05	41	45.2	2.7	26690*
long_hint_01	40	42.8	1.8	15060*
long_hint_02	28	30.2	1.1	14251*
long_hint_03	55	57.5	1.5	31694*
long_late_01	235	239.2	2.7	24811*
long_late_02	229	230.8	1.2	16966*
long_late_03	220	222.1	1.0	24035*
long_late_04	221	224.3	2.4	15406*
long_late_05	83	84.9	0.7	20434*

In Tables 6-8, the best result obtained over 25 runs is denoted by *Best*, the average result is denoted by *avg*, the standard deviation is denoted by σ, and the computational time of the best result is denoted by *Time* (where * is the time limit in INRC2010). For example, in Table 6, the PB-LS achieves a value of 57 as the best result (over 25 runs) for sprint_early_01 instance in 4 seconds, with an average of 58.7 and a standard deviation of 1.8. Tables 6, 7, and 8 illustrate that PB-LS obtains 43 out of 69 instances within the time limit (which marked as *) of INRC2010.

Results in Tables 3-5, and analysis in Figure 2 and Figure 3 confirm that the proposed approach can produce good quality solutions for the NRP compared to other existing approaches in the literature. In addition, results in Tables 6-8 show that the proposed approach runs in a good manner for all instances with instances of diversity. As a perpective to this work will be to apply the proposed approach in the context of remote sensing big data [41-44], to explore the context of case-based reasoning using 'hyper-heuristic' [45, 46], and to evaluate the effect of uncertainty in the process of search for nurse rostering problem [47-49].

5. CONCLUSION

In this paper, we proposed a population-based local search approach (PB-LS) for the nurse rostering problem. The population-based is motivated by a gravitational emulation local search algorithm to intensify the search space and by an MPCA-ARDA to diversify the search. A comparison is made between the performance of the proposed approach and performances of other exiting approaches in the literature over 69 datasets. In this paper, ten existing approaches are considered for comparison. Results indicate that the proposed approach produces a good quality solution compared to the existing approaches.

Our approach obtained 55 optimal solutions over 69 cases, and it is ranked first while comparing it to other approaches according to the number of optimal solutions. Additionally, results indicate that the proposed approach outperformed all the existing approaches while comparing the number of better solutions to the number of worse solutions. These results confirm the good performance of the proposed approach for solving the problem of nurse rostering.

As future work, we propose to use more case studies to test the performance of the proposed approach and to apply it in context of remote sensing big data. Additionally, in this paper, we use a single heuristic approach with different neighborhoods over a population of solutions; which is chosen based on gravity formula. Another challenging topic to be explored is the study of the case-based reasoning method to allow using of 'hyper-heuristic'. This will help to determine the better heuristic for a given population, and hence avoiding solving problems from scratch.

REFERENCES

[1] P. Brucker, et al., "Personnel scheduling: models and complexity," *European Journal of Operational Research*, vol. 210, no. 3, pp. 467-473, 2011.

[2] M. Dorigo and T. Stützle, "Ant colony optimization: overview and recent advances," Handbook of Metaheuristics, Springer, pp. 227-263, 2010.

[3] I. I. I. Bartholdi, "A guaranteed-accuracy round-off algorithm for cyclic scheduling and set covering," *Opererations Research*, vol. 29, no. 3, pp. 501-510, 1981.

[4] H. H. Millar and M. 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.

[5] Z. Lü and H. K. Hao, "Adaptive neighborhood search for nurse rostering," *European Journal of Operational Research*, vol. 218, no. 3, pp. 865-876, 2012.

[6] G. M. Jaradat, et al., "Hybrid Elitist-Ant System for Nurse-Rostering Problem," *Journal of King Saud University – Computer and Information Sciences*, vol. 31, no. 3, pp. 378-384 2019.

[7] M. Rajeswari, et al., "Directed Bee Colony Optimization Algorithm to Solve the Nurse Rostering Problem," *Computational Intelligence and Neuroscience*, vol. 2017, pp. 1-26, 2017.

[8] M. A. Awadallah, et al., "Hybridization of harmony search with hill climbing for highly constrained nurse rostering problem," *Neural Computing and Applications*, vol. 28, no. 3, pp. 463-482, 2017.

[9] H. G. Santos, et al., "Integer programming techniques for the nurse rostering problem," *Annals of Operations Research*, vol. 239, no. 1, pp. 225-251, 2016.

[10] M. A. Awadallah, et al., "A hybrid artificial bee colony for a nurse rostering problem," *Applied Soft Computing*, vol. 35, pp. 726-739, 2015.

[11] E. K. Burke and T. Curtois, "New approaches to nurse rostering benchmark instances," *European Journal of Operational Research*, vol. 237, no. 1, pp. 71-81, 2014.

[12] B. Bilgin, et al., "One hyper-heuristic approach to two timetabling problems in health care," *Journal of Heuristics*, vol. 18, no. 3, pp. 401-434, 2012.

[13] C. Valouxis, et al., "A systematic two phase approach for the nurse rostering problem," *European Journal of Operational Research*, vol. 219, no. 2, pp. 425-433, 2012.

[14] K. Nonobe, "INRC2010: An approach using a general constraint optimization solver," *The First International Nurse Rostering Competition (PATAT-2010)*, 2010. [Online]. Available: http://www.kuleuven-kortrijk.be/nrpcompetition.

[15] J. F. Bard and H. W. Purnomo, "Preference scheduling for nurses using column generation," *European Journal of Operational Research*, vol. 164, no. 2, pp. 510-534, 2005.

[16] H. G. Santos, et al., "Integer programming techniques for the nurse rostering problem," *Annals of Operations Research*, vol. 239, no. 1, pp. 225-251, 2016.

[17] S. Ceschia, et al., "Solving the INRC-II nurse rostering problemby simulated annealing based on large-scale neighborhoods," in *Proceedings of the 12th International Confenference on Practice and Theory of Automated Timetabling (PATAT-2018)*, 2018.

[18] L. Antoine, et al., "A rotation-based branch-and-price approach for thenurse scheduling problem," *Mathematical Programming Computation*, pp. 1-34, 2019.

[19] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: overview and conceptual comparison," *ACM Computing Surveys*, vol. 35, no. 3, pp. 268-308, 2003.

[20] D. yvaz, et al., "Performance evaluation of evolutionary heuristics in dynamic environments," *Applied Intelligence*, vol. 37, no. 1, pp. 130-144, 2012.

[21] H. R. Cheshmehgaz, et al., "Effective local evolutionary searches distributed on an island model solving bi-objective optimization problems," *Applied Intelligence*, vol. 38, no. 3, pp. 331-356, 2013.

[22] M. K. Alsmadi, "Query-sensitive similarity measure for content-based image retrieval using meta-heuristic algorithm," *Journal of King Saud University-Computer and Information Sciences*, vol. 30, no. 3, pp. 373-381, 2018.

[23] M. K. Alsmadi, "Content-Based Image Retrieval Using Color, Shape and Texture Descriptors and Features," *Arabian Journal for Science and Engineering*, vol. 45, pp. 3317-3330, 2020.

[24] M. K. Alsmadi, et al., "Robust feature extraction methods for general fish classification," *International Journal of Electrical & Computer Engineering*, vol. 9, no. 6, pp. 5192-5204, 2019.

[25] M. K. Alsmadi, "Hybrid Genetic Algorithm with Tabu Search with Back-Propagation Algorithm for Fish Classification: Determining the Appropriate Feature Set," *International Journal of Applied Engineering Research*, vol. 14, no. 23, pp. 4387-4396, 2019.
[26] M. K. Alsmadi, "An efficient similarity measure for content based image retrieval using memetic algorithm," *Egyptian Journal of Basic and Applied Sciences*, vol. 4, no. 2, pp. 112-122, 2017.
[27] M. Alsmadi, "Facial Recognition under Expression Variations," *The International Arab Journal of Information Technology*, vol. 13, no. 1A, pp. 133-141, 2016.
[28] M. Alsmadi, et al., "A hybrid memetic algorithm with back-propagation classifier for fish classification based on robust features extraction from PLGF and shape measurements," *Information Technology Journal*, vol. 10, no. 5, pp. 944-954, 2011.
[29] M. Alsmadi, et al., "Fish Classification: Fish Classification Using Memetic Algorithms with Back Propagation Classifier," LAP Lambert Academic Publishing, 2012.
[30] K. Yousef, et al., "Applying the big bang-big crunch metaheuristic to large-sized operational problems," *The International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 3, pp. 2848-2502, 2020.
[31] A. Abuhamdah, "Adaptive elitist-ant system for medical clustering problem," *The Journal of King Saud University - Computer and Information Sciences*, vol. 32, pp. 709-717, 2020.
[32] A. Abuhamdah, "Adaptive Acceptance Criterion (AAC) algorithm for Optimization Problems," *The Journal of Computer Science*, vol. 11, no. 4, pp. 675-691, 2015.
[33] A. Abuhamdah, et al., "Adaptive Great Deluge (AGD) for Medical Clustering Problem," *The Int. J. Emerg. Sci. (IJES)*, vol. 4, no. 1, pp. 1-13, 2014.
[34] A. Abuhamdah, et al., "Hybridization Between Iterative Simulated Annealing and Modified Great Deluge for Medical Clustering Problems," *World of Computer Science and Information Technology Journal (WCSIT)*, vol. 2, no. 4, pp. 131-136, 2012.
[35] A. Abuhamdah and M. Ayob, "Adaptive Randomized Descent Algorithm using Round Robin for Solving Course Timetabling Problems," *International Conference on Intelligent Systems Design and Applications (ISDA) IEEE*, pp. 1201-1206, 2010.
[36] A. Abuhamdah, et al., "Population based local search for university course timetabling problems," *Applied Intelligence*, vol. 40, no. 1, pp. 44-53, 2013.
[37] S. Haspeslagh, et al., "The first international nurse rostering competition," *Annals of Operations Research*, vol. 218, no. 1, pp. 221-236, 2014.
[38] S. Ceschia, et al., "The second international nurse rostering competition," *Annals of Operations Research*, vol. 274, pp. 171-186, 2019.
[39] B. L. Webster, "Solving combinatorial optimization problems using a new algorithm based on gravitational attraction," PhD thesis, College of Engineering at Florida Institute of Technology, 2004.
[40] A. Abuhamdah and M. Ayob, "MPCA-ARDA for solving course timetabling problems," in *3rd conference on data mining and optimization (DMO)*, New York, pp. 171-177, 2011.
[41] I. Chebbi, et al., "Big data: Concepts, challenges and applications," in *Computational collective intelligence*, Springer, Cham, pp. 638-647, 2015.
[42] I. Chebbi, et al. "A comparison of big remote sensing data processing with Hadoop MapReduce and Spark," in *4th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), IEEE*, pp. 1-4, 2018.
[43] W. Boulila, et al., "A novel decision support system for the interpretation of remote sensing big data," *Earth Science Informatics*, vol. 11, no. 1, pp. 31-45, 2018.
[44] W. Boulila, "A top-down approach for semantic segmentation of big remote sensing images," *Earth Science Informatics*, vol. 12, no. 3, pp. 295-306, 2019.
[45] I. R. Farah, et al., "Interpretation of multi-sensor remote sensing images: multi-approach fusion of uncertain information," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 12, pp. 4142-4152, 2008.
[46] I. R. Farah, et al., "Multiapproach system based on fusion of multispectral images for land-cover classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 12, pp. 4153-4161, 2018.
[47] A. Ferchichi, et al., "Reducing uncertainties in land cover change models using sensitivity analysis," *Knowledge and Information Systems*, vol. 55, no .3, pp. 719-740, 2018.
[48] W. Boulila, al., "Combining Decision Fusion and Uncertainty Propagation to Improve Land Cover Change Prediction in Satellite Image Databases," *The Journal of Multimedia Processing and Technologies*, vol. 2, no. 3, pp. 127-139, 2011.
[49] W. Boulila, et al., "Sensitivity analysis approach to model epistemic and aleatory imperfection: Application to Land Cover Change prediction model," *Journal of computational science*, vol. 23, pp. 58-70, 2017.