



Artificial Bee Colony Algorithm with Nelder–Mead Method to Solve Nurse Scheduling Problem

Rajeswari Muniyan ¹, Rajakumar Ramalingam ², Sultan S. Alshamrani ³, Durgaprasad Gangodkar ⁴, Ankur Dumka ^{4,5}, Rajesh Singh ^{6,7}, Anita Gehlot ^{6,7} and Mamoon Rashid ^{8,*}

- ¹ Department of Information Technology, Sri Manakula Vinayagar Engineering College, Pondicherry 605107, India; raji.yadaw1990@gmail.com
- ² Department of Computer Science and Technology, Madanapalle Institute of Technology & Science, Madanapalle 517325, India; drrajakumarr@mits.ac.in
- ³ Department of Information Technology, College of Computer and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia; susamash@tu.edu.sa
- ⁴ Department of Computer Science & Engineering, Graphic Era Deemed to Be University, Dehradun 248002, India; dr.gangodkar@geu.ac.in (D.G.); ankurdumka@geu.ac.in (A.D.)
- ⁵ Department of Computer Science and Engineering, Women Institute of Technology, Dehradun 248007, India
 ⁶ Uttaranchal Institute of Technology, Uttaranchal University, Dehradun 248007, India;
- rajeshsingh@uttaranchaluniversity.ac.in (R.S.); anita.ri@uttaranchaluniversity.ac.in (A.G.)
- ⁷ Department of Project Management, Universidad Internacional Iberoamericana, Campeche 24560, Mexico
- ⁸ Department of Computer Engineering, Faculty of Science and Technology, Vishwakarma University, Pune 411048, India
- Correspondence: mamoon.rashid@vupune.ac.in; Tel.: +91-7814346505

Abstract: The nurse scheduling problem (NSP) is an NP-Hard combinatorial optimization scheduling problem that allocates a set of shifts to the group of nurses concerning the schedule period subject to the constraints. The objective of the NSP is to create a schedule that satisfies both hard and soft constraints suggested by the healthcare management. This work explores the meta-heuristic approach to an artificial bee colony algorithm with the Nelder–Mead method (NM-ABC) to perform efficient nurse scheduling. Nelder–Mead (NM) method is used as a local search in the onlooker bee phase of ABC to enhance the intensification process of ABC. Thus, the author proposed an improvised solution strategy at the onlooker bee phase with the benefits of the NM method. The proposed algorithm NM-ABC is evaluated using the standard dataset NSPLib, and the experiments are performed on various-sized NSP instances. The performance of the NM-ABC is measured using eight performance metrics: best time, standard deviation, least error rate, success percentage, cost reduction, gap, and feasibility analysis. The results of our experiment reveal that the proposed NM-ABC algorithm attains highly significant achievements compared to other existing algorithms. The cost of our algorithm is reduced by 0.66%, and the gap percentage to move towards the optimum value is 94.30%. Instances have been successfully solved to obtain the best deal with the known optimal value recorded in NSPLib.

Keywords: meta-heuristic approach; evolutionary algorithm; nurse scheduling problem; artificial bee colony algorithm; Nelder–Mead method

MSC: 68Q17; 68Q30; 68Q87

1. Introduction

In a hospital, various operations are performed; nurse rostering is a resource allocation problem. Every day, work is divided into three periods: day shift, noon shift, and night shift. The process consists of allocating workload to the nurses periodically by considering hospital terms, namely constraints and requirements, for a scheduling period of one month. Constraints are classified as hard and soft constraints; hard constraints are needed to be satisfied when allocating the roster. The soft constraints are considered when the



Citation: Muniyan, R.; Ramalingam, R.; Alshamrani, S.S.; Gangodkar, D.; Dumka, A.; Singh, R.; Gehlot, A.; Rashid, M. Artificial Bee Colony Algorithm with Nelder–Mead Method to Solve Nurse Scheduling Problem. *Mathematics* **2022**, *10*, 2576. https://doi.org/10.3390/ math10152576

Academic Editor: Cristian Ramírez Atencia

Received: 17 June 2022 Accepted: 19 July 2022 Published: 25 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). prerequisite is desirable but not obligatory. The soft constraint will estimate the quality of the roster; these constraints are considered as much as possible, but violating soft constraints leads to a penalty, while the roster is still considered feasible. Each nurse is allocated to a specific number of shifts with various constraints. Each shift is accomplished by the group of nurses based on their preferences and skills required for the particular operation. The NSP is a multi-objective NP-hard combinatorial problem [1–4].

Nurse scheduling is allocating nurses with different services and bonds in a subject satisfying the constraints to reduce the overall cost of the scheduling problem. The NSP is the placement of nurses with different skill patterns into shifts pattern in a subject satisfying the constraints. This problem focuses on generating a roster and representing slot allotment for the nurses to shift. The obtained roster solution should persuade hospital regulations, nurse preferences, and requests listed in terms of soft constraints. Two methods can enable the scheduling of nurses: cyclic rostering and non-cyclic rostering. In cyclic rostering, each nurse is assigned to a shift for the planning epoch. The same cyclic method is repeated without modifications in the shift pattern for consecutive days. In this approach, the workload among the nurses can easily be distributed among them within a stipulated time [5,6]. In non-cyclic rostering, the shift pattern allotted to the nurse keeps changing for various periods. The new shift pattern is scheduled based on the nurse's preferences, requests, and constraints. The non-cyclic approach is more advantageous than cyclic rostering because updating the roster will cause the entire schedule to modify. In contrast, in the non-cyclic process, only the pattern part needs to be changed. The non-cyclic approach of rostering will effectively inscribe all constraints, preferences, and requirements required [7].

The objective is to increase the nurses' priority and diminish the total penalty cost from desecrations of the soft constraints. Most researchers have considered only a single goal, and only a few seek multi-objectives to solve NSP [8–10]. NSP is the assignment of nurses to the shifts, and the main objective is to assign nurses by satisfying the hospital regulations. The NSP is solved using heuristic, meta-heuristic, and mathematical programming models [11–13].

The automatic planning of NSP [14] is to generate a roster for working nurses and rest days for a particular period concerning a set of constraints. The head nurse manually commences this roster; scheduling nurses for three shifts for one month hardly takes six to eight hours. This roster may be preliminary since the changes occur based on the nurse's opinion. Some researchers developed approaches to solving NSP. Aickelin and Dowsland [15] used a genetic algorithm framework with Tabu search as a local search to exploit the solution. Burke et al. used simulated annealing multi-objective approach to solve NSP and generate a key in the Pareto front [16].

The mathematical programming model is the first approach for solving the NSP; although mathematical programming methods ensure that they will provide the optimum solution, they fail to do so within a rational amount of time when the solution search space is vast. This requires a constraint-based programming method to solve complicated and soft constraints [17]. A heuristic approach is ideal for solving the problem with benign conditions and functionality. The heuristic-based approaches, such as the local search approaches of simulated annealing [18], iterated local search [19], and Tabu search [8], are combined with other methods to solve the NSP effectively. The meta-heuristic approach is implemented over low-level heuristics to improve their performance. The hybrid system combines the heuristic method with any deterministic process, such as a mathematical programming model [20]. The population-based way is a type of heuristic such as a genetic algorithm [21], scatter search [22], particle search [23], and ant colony optimization [24]. These approaches generate a population of solutions from which a better solution can be achieved.

Berrada et al. proposed a lexico-dominance technique to solve NSP with several soft constraints. The soft constraints are prefixed with priority values and ordered to obtain the best solution [25]. Burke et al. applied weight values to the soft constraints, which are determined by the hospital regulations [26]. The conditions are in order based on the weight values, and the best solution is chosen. Nikola and Petrovic introduced a bee

colony optimization algorithm to solve NSP, and unscheduled shifts are assigned to the nurse constructively, leading to higher penalties [27,28]. Meng Yang et al. introduced a multi-objective artificial bee colony algorithm to solve the multi-objective home healthcare routing and scheduling problem. Based on the enormous neighborhood search, a heuristic solution update strategy has been proposed to trade off the search balance and achieve the workload balance in the scheduling [29].

In this research work, the authors introduced the artificial bee colony algorithm with the Nelder–Mead method (NM-ABC) to solve nurse scheduling problems. The local search mechanism is incorporated in the onlooker bee phase of ABC to obtain an optimal solution for NSP. ABC algorithm produced good results in solving optimization problems and improved the efficiency of the global search, memory management, and search improvement process [30–37]. However, ABC follows the same neighborhood search strategy in the onlooker and employee bee phase. Improving neighborhood search on quality individuals enhances the probability of convergence towards global optimum compared to a random process. Thus, this paper introduces the Nelder–Mead Local search in the onlooker bee phase to improve the neighborhood search.

The main contributions of this work are illustrated as follows:

- 1. A hybrid meta-heuristic algorithm, namely artificial bee colony optimization with Nelder–Mead Method, is proposed.
- 2. The search capability of ABC is enriched with the aid of the Nelder–Mead method, which consists of search strategies such as midpoint, reflection, expansion, contraction, and shrinkage processes. These search strategies enhance the balance between exploration and exploitation.
- NM-ABC is implemented and tested on the nurse scheduling problem (NSPLib).
- The performance of NM-ABC is compared with that of some classical optimization algorithms.

The rest of the paper is structured as follows: Section 2 illustrates the scientific formulation of NSP. In Section 3, the proposed artificial bee colony algorithm with NM is discussed. Section 4 shows the applicability of NM-ABC to solve NSP with experimental analysis. Section 5 deliberates a detailed analysis of the performance outcome of NM-ABC to solve NSP. Finally, Section 6 concludes the research work.

2. Nurse Scheduling Problem

The NSP can be described as allocating a set of nurses to a group of shifts for a given time. The constraints are organized by hospital regulations, nurse preferences, nurse requests, and working practices. Generally, NSP is divided into two different types of constraints: they are hard and soft constraints. Hard constraints are the regulations that must be gratified to achieve a feasible solution. The roster pattern generated should satisfy all hard constraints; generally, the general hard constraint is allocating shifts to the restricted number of nurses. Soft constraints are desirable and not obligatory but must be satisfied as much as possible. The soft constraints will determine the quality of the roster formed. The violation of any soft constraints leads to the penalty of the solution. The general soft constraint in NSP is to balance the workload among all nurses and usage of nurse resources efficiently [38,39]. The visual representation of NSP is illustrated in Figure 1.

The problem of NSP consists of a set of nurses $N = \{1, 2, ..., i\}$ that is allocated a set of shifts $S = \{1, 2, ..., j\}$ for the scheduled period $D = \{1, 2, ..., k\}$ in days. The shift pattern *r* consists of the allocation of shifts for the particular nurse over the expected period for the 0/1 matrix. The main objective of the NSP is to identify a possible solution such that the total cost is minimized. The solution representation of the NSP is given as

$$X_{j,d,r} = \begin{cases} 1, \text{ pattern covers shift } j \text{ on } day d \\ 0, & \text{otherwise} \end{cases}$$
(1)



Figure 1. Visual Representation of Nurse Scheduling Problem.

The nurse preference is the wish expressed by a particular nurse to work on a distinct shift for a specific day. The penalty cost is added to the solution if the nurse's preference is not achieved. The preference $\cot C_{i,j,d}$ is the wish of the particular nurse *i* to work on shift *j* on a particular day *d*. The total preference cost of the shift pattern *r* for the scheduled period can be calculated as

$$P_{i,r} = \sum_{j=1}^{S} \sum_{d=1}^{D} X_{j,d,r} * C_{i,j,d}$$
⁽²⁾

The objective function of the NSP can be formulated as

$$Minimize \sum_{i=1}^{N} \sum_{r=1}^{\delta_i} X_{j,r} * P_{i,r}$$
(3)

where δ_i is the set of a possible shift pattern for the nurse *i* to work on the scheduled period. Subject to

$$\sum_{r=1}^{\delta_i} X_{j,r} = 1 \tag{4}$$

The feasible solution is for all nurses available in the schedule, and this constraint specifies that precisely one shift pattern is allocated to every nurse in the hospital. The smallest number $M_{j,d}$ of nurses essential for the shift *j* on the scheduled day *d* can be restricted by using Equation (5).

$$\sum_{i=1}^{N} \sum_{r=1}^{\delta_i} X_{j,d,r} \ge M_{j,d}$$

$$\tag{5}$$

$$X_{i,r} \in \{0,1\}$$
 (6)

The Equations (4)–(6) are the coverage constraints considered in solving the NSP.

All the hard constraints in the NSP are mandatory and should be considered when designing the schedule pattern. The soft constraints are included in the objective function of increasing the quality of the solution. Soft restrictions are not mandatory and enhance the quality of the schedule. Violation of soft constraints leads to penalty cost, and the hospital management suggests the penalty cost. Some of the soft constraints are listed in this work, and the soft constraints (SC) are as follows

SC1:

In this constraint, the nurse has restricted work on a specific day. The constraint can be evaluated as

$$\sum_{i=1}^{N} X_{i,d} \le 0 \tag{7}$$

SC2:

The nurse can request to work on a specific day. The constraint can be calculated as

$$\sum_{i=1}^{N} X_{i,d} \ge 1 \tag{8}$$

SC3:

The nurse can request not to work on a specific shift. The constraint can be

$$\sum_{i=1}^{N} X_{i,j} \le 0 \tag{9}$$

SC4:

The nurse can request to work on a specific shift. The constraint can be determined as

$$\sum_{i=1}^{N} X_{i,j} \ge 1 \tag{10}$$

SC5:

No single work shift between two days off. The constraint can be calculated as

$$X_{i,(j-1)} - X_{i,j} + X_{i,(j+1)} \ge 0$$
(11)

SC6:

The nurses are not allowed to work more than three consecutive days. The constraint can be

$$X_{i,2j} - 3 \ge 0 \tag{12}$$

3. Proposed Algorithm

3.1. Artificial Bee Colony Algorithm

Karaboga and Bahriye developed an artificial bee colony algorithm (ABC), inspired by honey bees' natural behavior. The intelligent behavior of honey bees helps to find the near-optimal solution for the optimization problem [32]. ABC is the population-based algorithm and consists of three groups of honey bees: employed bee, onlooker bee, and scout bee. The colony consists of an equal number of employed bees and onlooker bees. Each solution in the population is held by one employed bee. The employed search for the food source and share the direction of the food source with onlooker bees through waggle dance. Based on the probability calculation, the higher-quality food sources are selected by the onlooker bee phase, and the bees continue further searches. The low-quality food sources are rejected, and employed bees are converted to scout bees. The scout bee will search for a new food source or food position.

3.1.1. Initialization

The initialization of the population is created with food source FS for *n*-dimensional vectors. The population of the solution is represented as $X_i = \{x_{i,1}, x_{i,2}, ..., x_{i,n}\}$. The food source in the population is generated using Equation (13).

$$x_{i,j} = x_{min,j} + rand \ (0,1) * (x_{max,j} - x_{min,j}) \tag{13}$$

The food sources are randomly distributed to employed bees. The objective calculation for the solution is evaluated accordingly.

3.1.2. Employed Bee Phase

In the employed bee phase, the candidate solution is generated, and the position of the food sources is monitored. The candidate solution can be developed using

$$v_{i,j} = x_{i,j} + \mathcal{O}_{i,j}(x_{i,j} - x_{k,j})$$
(14)

where j = 1, 2, ..., S, and we choose the value of k as different from I, k = 1, 2, ..., FS, and the value of \emptyset ranges (-1, 1). The greedy selection is made between v_i and x_i based on the fitness calculation. If the fitness value of v_i is greater than x_i , the solution x_i is replaced by v_i .

3.1.3. Probability Calculation

After fitness calculation, the employed bees share the direction of the food source with onlooker bees. The onlooker bees evaluate the fitness of the employed bee's solution using the probability value p_i . The solution's probability and fitness calculation are shown in detail in Algorithm 1.

Algo	rithm 1: Probability calculation
1:	For $i = 1, 2,, FS$, do
2:	Calculate probability values P_{ij} for the solution $v_{i,j}$
3:	$P_i = rac{fit_i}{\sum_{j=1}^{FP} fit_j}$
4:	$fit_i = \begin{cases} \frac{1}{1+f_i}, & f_i \ge 0 \end{cases}$
	$1 + abs(f_i), \qquad f_i < 0$
5:	End For

3.1.4. Onlooker Bee Phase

Based on the probability value p_i , Each onlooker bee randomly chooses the food source x_i with probability p_i . Onlooker bee performs modification on x_i using Equation (14). To select the best solution among x_i and v_i , the greedy method is used, which is similar to the employed bee phase.

3.1.5. Scout Bee Phase

After employed and onlooker bees search, the food source is abandoned if the solution cannot be improved and exhausted for the predefined number of iterations. The corresponding employed bee develops a scout bee and explores new food sources using Equation (13).

3.2. Nelder-Mead Method

The NM method is used to find the local minimum for the given function and is represented as a simple triangle with three vertices. The worst vertex is found among the triangle, rejected for the next iteration, and replaced with a new vertex. The search continues toward the best solution in the triangle sequence, reducing the triangle size. At last, the vertex with a minimum point is chosen as the best solution.

Let f(x, y) be the function to be minimized; start with three vertices of a triangle. The fitness function is evaluated for all three points of the triangle. The vertices are ordered based on the fitness value: best (I), good (J), and worst (K) vertices are ordered. NM method works on four operations: reflection, expansion, contraction, and shrinkage.

3.2.1. Midpoint (M)

The midpoint between the line joining vertices is calculated for the first two best and good vertices using

$$M = \frac{I+J}{2} \tag{15}$$

3.2.2. Reflection (R)

The function increases towards the side of the triangle when we move from the worst vertex to the best vertex and decreases when moving from the worst to the excellent vertex. The test point reflection point *R* is chosen along the side of \overline{IJ} . To find *R*, calculate the midpoint of the best and good vertex because the best solution is away from the worst

vertex. Draw a line joining *K* and *M* of length *d*. Place *R* extending at a distance *d*. The formula to calculate *R* is

R

$$=2M-K \tag{16}$$

3.2.3. Expansion (E)

The fitness calculation at vertex R is calculated, and if it is less than K, then the search is moved towards a minimum value. Now, extend the line segment through the vertex R to E by the distance d. If the fitness value at vertex E is better than R, then it is towards the minimum value. The formula to calculate E is given by

$$E = 2R - M \tag{17}$$

3.2.4. Contraction (C)

When the fitness value of *R* and *K* are small, another vertex in the triangle is needed to continue the process. Then, contraction towards midpoint without replacement is performed. The contraction points C_1 and C_2 are drawn along the line joining \overline{KM} and \overline{MR} for the length of $\frac{d}{2}$. The formula to calculate *C* is

$$C = R + M \tag{18}$$

3.2.5. Shrinkage (S)

The fitness value at *C* is calculated, and if it is not less than *K*, then the vertices *J* and *K* should be shrunk towards the best vertex *I*. The vertex *J* is replaced with *M*, and *K* is replaced with *S*. The point *S* is the midpoint of *I* and *K*. The process is continued until the minimum value is found. The detailed description of the flow of the NM method is shown in Algorithm 2.

Alg	orithm 2: Nelder–Mead Method
1:	Produce new food source v_i using modified Nelder–Mead Method
2:	Let v_i denote list of vertices
3:	ζ , μ , λ , and ζ are the constants of likeness, extension, shrinkage, and contraction
4:	f is the fitness method to be reduced
5:	<i>For</i> $i = 1, 2,, n + 1$ vertices, <i>do</i>
6:	Order the vertices from deepest fitness function $f(v_1)$ to maximum fitness function
	$f(\llbracket v bracket_{(n+1)})$
7:	$f(v_1) \le f(v_2) \le \dots \le f(v_{n+1})$
8:	Calculate midpoint for best two vertices
9:	$v_m = \sum \frac{v_i}{n}$, where $i = 1, 2, \dots, n$
10:	Calculate reflection point v_r
11:	$v_r = v_m + \mathfrak{l}(v_m - v_{n+1})$
12:	if $f(v_1) \leq f(v_r) \leq f(v_n)$ then
13:	$v_n \leftarrow v_r$ and go to stopping criteria
14:	End if
15:	Calculate expansion point v_e
16:	if $f(v_r) \leq f(v_1)$ then
17:	$v_n \leftarrow v_e$ and go to stopping criteria
18:	End if
19:	if $f(v_e) < f(v_r)$ then
20:	$v_n \leftarrow v_e$ and go to stopping criteria
21:	else
22:	$v_n \leftarrow v_r$ and go to stopping criteria
23:	End if
24:	Calculate contraction point v_c
25:	if $f(v_n) \leq f(v_r) \leq f(v_{n+1})$ then
26:	Compute outside contraction
27:	$v_c = \lambda v_{n+1} + (1 - \lambda) v_m.$
28:	Ena tj

29:	if $f(v_r) \ge f(v_{n+1})$ then
30:	Compute inside contraction
31:	$v_c = \lambda v_{n+1} + (1-\lambda) v_m.$
32:	End if
33:	if $f(v_r) \ge f(v_n)$ then
34:	Shrinkage is done between v_m and the best vertex among v_r and v_{n+1} .
35:	End if
36:	if $f(v_c) < f(v_r)$ then
37:	$v_n \leftarrow v_c$ and go to Stopping criteria
38:	<i>else</i> go to Shrinkage phase
39:	End if
40:	if $f(v_c) \ge f(v_{n+1})$ then
41:	$v_{n+1} \leftarrow v_c$ and go to Stopping criteria
42:	<i>else</i> go to the Shrinkage phase
43:	End if
44:	Calculate Shrinkage
45:	Shrink close the best individual with new apices
46:	$v_i = \zeta v_i + v_1(1 - \zeta)$, where $i = 2,, n + 1$
47:	End for
18.	Determine the new vertices of the simplex thus formed based on their fitness and continue
1 0.	with the process of the reflection phase

3.3. Nelder–Mead Method-Based ABC (NM-ABC)

ABC algorithm is lithe to improve and progress; the intricacy of the algorithm is reduced since it uses fewer parameters. The improved search ability aids in attaining optimal solutions with less computational time and increased convergence speed. ABC is good at examination but fails in exploitation [40]. An improved local search algorithm is incorporated in ABC to tradeoff the search [41].

The Nelder–Mead method is the famous local search algorithm, and it is simple and efficient. It is also proficient at embedding in other global search algorithms. However, it is entrapped in a local optimum solution. Thus, it is poor at exploration and has low convergence towards the initial position. Nelder–Mead (NM) method is good at the exploitation process but poor in the exploration process. This paper uses the NM method in ABC to improve the exploitation. The detailed description of the proposed algorithm's pseudo code is shown in Tables 1–3. The workflow of the proposed NM-ABC is shown in Figure 2. The proposed NM-AMC algorithm is presented in Algorithm 3.

Table 1. Instances Taken from dataset with the values of Nurse, Day, and Shift.

Case	Туре	Instances	Nurse	Day	Shift
1	N25	1, 7, 12, 19, 25	25	7	4
2	N25	2, 5, 9, 15, 27	25	7	4
3	N25	1, 3, 16, 27, 35	25	7	4
4	N25	5, 10, 25, 38, 41	25	7	4
5	N25	7, 11, 30, 42, 47	25	7	4
6	N50	1, 4, 12, 26, 29	50	7	4
7	N50	3, 6,12, 26, 36	50	7	4
8	N50	4, 9, 15, 40, 47	50	7	4
9	N50	5, 10, 23, 29, 40	50	7	4
10	N50	6,14, 20, 32, 41	50	7	4
11	N60	2, 8, 14, 20, 32	60	28	4
12	N60	3, 12, 19, 23, 34	60	28	4
13	N60	1, 4, 19, 29, 40	60	28	4
14	N60	5, 9, 15, 30, 43	60	28	4
15	N60	6, 15, 26, 35, 44	60	28	4

Туре	Method	Parameters and Values
M ₁	Multi-Assignment Problem-based Algorithm (MAPA) [3]	Number of Iterations—1000, Penalty Violation Value—100
M ₂	Hybrid Artificial Bee Colony Algorithm (HABC) [2]	Limit—100, Spin Track—10, Number of Population—100, Number of Iterations—100
M ₃	Bee Colony Optimization Algorithm (BCO) [27]	Knowledge Base (b)—2 Number of Population—100, Number of Iterations—100
M_4	Hybrid Elitist–Ant System (HEAS) [12]	Population Size—100, Number of Iterations—1000, Pheromone Initial Values—0.01 Evaporation Rate—0.25
M ₅	Harmony Search-Based Hyper-Heuristic Algorithm (HSHH) [1]	Number of Population—100, Number of Iterations—100
Proposed	Artificial Bee Colony with Nelder–Mead (NM-ABC)	Number of Bees—100, Number of Iterations—1000, Maximum Runs—20, Reflection Coefficient— $\alpha > 0$ Expansion Coefficient— $\gamma > 1$ Contraction Coefficient— $0 > \beta > 1$ Shrinkage Coefficient— $0 < \delta < 1$

 Table 2. Parameters setting of compared methods.

Table 3. Summary of the Best result obtained by proposed Artificial bee colony with Nelder Mead method (NM-ABC) and competitor methods.

Case	Туре	Instance	Optimal Value	NM-ABC	M1	M ₂	M ₃	M_4	M ₅
		1	307	307	307	307	307	306	307
		7	291	287	290	292	290	292	292
C-1	N25	12	296	296	297	296	296	296	296
		19	302	302	302	302	303	302	302
		25	308	307	307	306	307	308	308
		2	274	269	277	279	276	273	273
		5	303	303	303	302	302	302	301
C-2	N25	9	276	276	276	276	276	276	275
		15	296	289	299	300	299	296	300
		27	293	293	293	293	293	293	293
		1	333	331	333	339	337	332	332
	N25	3	315	315	315	315	315	317	315
C-3		16	323	325	322	323	326	326	323
		27	318	316	318	318	319	318	317
		35	333	327	330	330	331	333	333
		5	313	285	301	319	310	312	316
		10	284	273	275	288	277	281	279
C-4	N25	25	308	294	303	300	305	306	312
		38	294	294	294	294	294	293	299
		41	296	296	296	296	296	299	296
		7	293	289	299	299	293	293	296
		11	299	299	299	299	299	298	299
C-5	N25	30	311	309	310	319	315	311	310
		42	283	283	287	283	283	283	283
		47	310	309	309	315	309	309	309

Table 3. Cont.

$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	Case	Туре	Instance	Optimal Value	NM-ABC	M_1	M_2	M ₃	\mathbf{M}_{4}	M_5
$ \begin{array}{c c} \begin{array}{c c} -6 \\ -8 \\ -8 \\ -8 \\ -8 \\ -9 \\ -9 \\ -9 \\ -9$			1	575	569	579	577	573	575	577
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$			4	641	640	641	640	641	640	649
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	C-6	N50	12	575	571	578	580	575	575	575
$\begin{array}{c c c c c c c c c c c c c c c c c c c $	CO	1000	26	566	566	569	566	566	566	565
$\begin{array}{c c} -1 & 2 & 3 & 59 & 512 & 590 & 597 & 595 & 601 & 599 \\ \hline C.7 & N50 & 12 & 606 & 600 & 609 & 603 & 608 & 606 & 612 \\ 26 & 579 & 574 & 578 & 578 & 579 & 580 & 581 \\ 36 & 630 & 630 & 630 & 630 & 630 & 630 & 630 \\ \hline C.8 & N50 & 15 & 580 & 573 & 577 & 589 & 580 & 583 & 585 \\ 40 & 562 & 562 & 562 & 562 & 562 & 562 \\ 47 & 562 & 562 & 562 & 562 & 562 & 562 \\ 47 & 562 & 562 & 572 & 562 & 561 & 565 & 562 \\ 47 & 562 & 562 & 572 & 562 & 561 & 565 & 562 \\ \hline C.9 & N60 & 23 & 3476 & 3350 & 3393 & 3370 & 3362 & 3362 & 3368 & 3372 \\ \hline C.9 & N60 & 23 & 3476 & 3350 & 3393 & 3399 & 3394 & 3475 & 3477 \\ 29 & 3061 & 3025 & 3069 & 3061 & 3069 & 3061 & 3060 \\ 40 & 2786 & 2786 & 2786 & 2786 & 2786 & 2786 & 2786 \\ \hline C.10 & N60 & 21 & 3414 & 3341 & 3341 & 3397 & 3393 & 3399 & 3394 & 3440 \\ 41 & 3394 & 3390 & 3393 & 3399 & 3399 & 3394 & 3440 \\ 41 & 3314 & 3504 & 3510 & 3520 & 3313 & 3314 & 3520 \\ \hline C.11 & N60 & 14 & 3703 & 3700 & 3870 & 3870 & 3870 & 3870 & 3870 \\ \hline C.12 & N60 & 12 & 3442 & 3441 & 3441 & 3441 & 3441 & 3442 \\ 32 & 3398 & 3398 & 3398 & 3398 & 3398 & 3398 & 3398 \\ \hline C.11 & N60 & 14 & 3703 & 3700 & 3700 & 3706 & 3708 & 3711 & 3705 \\ \hline C.12 & N60 & 19 & 2988 & 2988 & 2988 & 2988 & 2988 & 2988 & 2988 \\ \hline C.13 & N60 & 19 & 2988 & 2368 & 3298 & 3398 & 3398 & 3398 & 3491 \\ \hline A & 3197 & 3197 & 3197 & 3197 & 3197 \\ \hline C.13 & N60 & 19 & 3284 & 3243 & 3244 & 3244 & 3244 & 3249 & 3249 \\ \hline C.13 & N60 & 19 & 3126 & 3125 & 3114 & 3132 & 3142 & 3422 & 3432 \\ \hline C.14 & N60 & 13 & 3203 & 3275 & 3291 & 3299 & 2988 & 2988 \\ 298 & 2969 & 2988 &$			20	575	575	572	575	575	573	577
$\begin{array}{c c} \begin{array}{c} 3 & 590 & 512 & 590 & 597 & 595 & 601 & 599 \\ \hline & 571 & 571 & 571 & 571 & 571 & 571 \\ & 571 & 571 & 571 & 571 & 571 & 571 \\ & 571 & 571 & 571 & 571 & 571 & 571 \\ & 56 & 630 & 630 & 630 & 630 & 630 & 630 \\ \hline & 36 & 630 & 630 & 630 & 630 & 630 & 630 \\ \hline & 36 & 630 & 573 & 577 & 578 & 578 & 578 \\ & 9 & 571 & 571 & 571 & 571 & 571 & 571 \\ & 571 & 571 & 571 & 571 & 571 & 571 \\ & 571 & 571 & 571 & 571 & 571 & 571 \\ & 571 & 571 & 571 & 571 & 571 & 571 \\ & 571 & 571 & 571 & 571 & 571 & 571 \\ & 571 & 571 & 571 & 571 & 571 & 571 \\ & 47 & 562 & 562 & 562 & 562 & 562 \\ \hline & 47 & 562 & 562 & 562 & 562 & 562 \\ \hline & 47 & 562 & 562 & 572 & 561 & 565 & 562 \\ \hline & 10 & 3114 & 3107 & 3112 & 3119 & 3117 & 3114 & 3123 \\ \hline & 10 & 3114 & 3107 & 312 & 3119 & 3117 & 3114 & 3123 \\ \hline & 0 & 2786 & 2786 & 2786 & 2786 & 2786 & 2786 & 2786 \\ \hline & 2786 & 2786 & 2786 & 2786 & 2786 & 2786 & 2786 & 2786 \\ \hline & 2786 & 2786 & 2786 & 2786 & 2786 & 2786 & 2786 \\ \hline & 14 & 3394 & 3390 & 3399 & 3399 & 3399 & 3394 & 3400 \\ \hline & 14 & 3703 & 3700 & 3700 & 3370 & 3370 & 3370 & 3370 \\ \hline & 32 & 3398 & 3398 & 3398 & 3398 & 3398 & 3398 & 3398 \\ \hline & 14 & 3703 & 3700 & 3700 & 3700 & 3370 & 3370 & 3370 \\ \hline & 32 & 3642 & 3639 & 3398 & 3398 & 3398 & 3398 & 3398 \\ \hline & 1 & 324 & 3243 & 3244 & 3244 & 3244 & 3244 \\ \hline & 2 & 284 & 2968 & 2967 & 2988 & 2988 & 2988 & 2988 \\ \hline & 2 & 3642 & 3649 & 3646 & 3646 & 3646 & 3646 & 3646 & 3646 \\ \hline & 3 & 3 & 3721 & 3720 & 3721 & 2725 & 2725 & 2725 \\ \hline & 1 & 860 & 14 & 3703 & 3700 & 3700 & 3700 & 3708 & 3701 & 370 & 3197 & 3197 & 3197 \\ \hline & 1 & 3244 & 3243 & 3244 & 3244 & 3244 & 3244 & 3244 & 3244 & 3244 & 3244 & 3244 & 3244 & 3244 & 3244 & 3249 & 3249 \\ \hline & 2 & 2 & 3642 & 3639 & 3639 & 3639 & 3641 & 3646 & 3644 & 3644 & 3641 & 3641 & 3641 & 3641 & 3641 & 3641 & 3641 & 3642 & 3641 & 3642 & 3641 & 3646 & 3646 & 36$			۷)	575	575	512	575	575	575	577
$\begin{array}{c c} \begin{array}{c} -r \\ -r $			3	590	512	590	597	595	601	599
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$			6	571	571	571	571	571	571	571
$\begin{array}{c c c c c c c c c c c c c c c c c c c $	C-7	N50	12	606	600	609	603	608	606	612
$\begin{array}{ c c c c c c c c c c c c c c c c c c c$			26	579	574	578	578	579	580	581
$\begin{array}{c c c c c c c c c c c c c c c c c c c $			36	630	630	630	630	630	630	630
$\begin{array}{c c} \begin{array}{c c} -8 & N50 & 15 & 571 $			4	644	640	642	645	642	648	645
$\begin{array}{cccccccccccccccccccccccccccccccccccc$			9	571	571	571	571	571	571	571
$\begin{array}{c c c c c c c c c c c c c c c c c c c $	C-8	N50	15	580	573	577	589	580	583	585
$\begin{array}{c c c c c c c c c c c c c c c c c c c $	60	1,000	40	562	562	565	562	562	562	561
$\begin{array}{c c c c c c c c c c c c c c c c c c c $			47	562	562	572	562	561	565	562
$\begin{array}{c c c c c c c c c c c c c c c c c c c $			F	22(2	2200	2270	22(2	22(2	22(0	2272
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$			5	3362	3299	3370	336Z	336Z	3368	3372
$ \begin{array}{c c} C-3 & \mathrm{N60} & 2.3 & 3476 & 3420 & 3479 & 3480 & 3475 & 3475 & 3477 \\ 29 & 3061 & 3025 & 3069 & 3061 & 3069 & 3061 & 3069 \\ 40 & 2786 & 2786 & 2786 & 2786 & 2786 & 2785 & 2786 \\ 14 & 3394 & 3390 & 3393 & 3399 & 3394 & 3400 \\ 20 & 3441 & 3441 & 3441 & 3441 & 3441 & 3441 & 3441 \\ 32 & 3398 & 3398 & 3397 & 3400 & 3398 & 3398 & 3401 \\ 32 & 3398 & 3398 & 3397 & 3400 & 3398 & 3398 & 3401 \\ 41 & 3514 & 3504 & 3510 & 3520 & 3513 & 3514 & 3520 \\ \hline & & & & & & & & & & & & & & & & & &$		2140	10	3114	3107	3112	3119	3117	3114	3123
$\begin{array}{c c c c c c c c c c c c c c c c c c c $	C-9	N60	23	3476	3450	3479	3480	3475	3475	3477
$\begin{array}{c c c c c c c c c c c c c c c c c c c $			29	3061	3025	3069	3061	3069	3061	3060
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$			40	2786	2786	2786	2786	2786	2785	2786
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$			6	2756	2756	2756	2756	2756	2756	2756
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$			14	3394	3390	3393	3399	3399	3394	3400
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	C-10	N60	20	3441	3441	3441	3441	3441	3440	3441
$\begin{array}{c c c c c c c c c c c c c c c c c c c $	010	1100	32	3398	3398	3397	3400	3398	3398	3401
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$			41	3514	3504	3510	3520	3513	3514	3520
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$			2	2970	2970	2970	2970	2970	2970	2870
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$			2	2502	3670	2508	2508	2508	2508	3670
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	0.11	NKO	0	5096 2702	3390	3396 2705	3396	3396	5596 2711	3396
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	C-11	N60	14	3703	3700	3705	3706	3708	3/11	3705
$\begin{array}{c c c c c c c c c c c c c c c c c c c $			20	3646	3646	3646	3646	3649	3646	3646
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$			32	3642	3639	3639	3641	3645	3652	3641
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$			3	2721	2720	2730	2721	2725	2729	2725
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$			12	2988	2976	2988	2988	2990	2998	2992
$\begin{array}{c c c c c c c c c c c c c c c c c c c $	C-12	N60	19	2988	2988	2988	2987	2988	2998	2988
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $			23	3432	3427	3431	3432	3432	3432	3431
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$			34	3197	3197	3196	3197	3197	3197	3197
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$			1	3244	3243	3244	3244	3244	3249	3249
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$			4	2988	2969	2989	2985	2996	2987	3001
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	C-13	N60	19	3136	3125	3141	3135	3139	3136	3139
$\begin{array}{c c c c c c c c c c c c c c c c c c c $	C 10	1000	29	3103	3100	3105	3103	3107	3102	3103
$\begin{array}{c c c c c c c c c c c c c c c c c c c $			40	2834	2834	2834	2834	2834	2834	2834
C-14 N60 15 3063 2972 3063 3056 3058 3069 3060 30 2935 2873 2928 2934 <td></td> <td></td> <td>F</td> <td>2202</td> <td>2075</td> <td>2201</td> <td>2200</td> <td>2200</td> <td>2200</td> <td>2200</td>			F	2202	2075	2201	2200	2200	2200	2200
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$			5	3293 2050	3275 2045	3291 2050	3299 2050	3299 2060	3300 2040	3299 2050
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	C 14	NKO	9	2959	2945	2959	2959	2969	2969	2959
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	C-14	IN60	15	3063	2972	3063	3036	3038	3069	3060
43 2963 2965 2962 2963 3035 3			30	2935	2873	2928	2934	2934	2934	2934
6303129973031304030303035303515338333833383338333833383338333833383C-15N60263969387539783979396939693979353496349234993496349635003500443475332834813479347234753473			43	2963	2965	2962	2963	2963	2963	2963
C-15N601533833383338333833383338333833383C-15N60263969387539783979396939693979353496349234993496349635003500443475332834813479347234753473			6	3031	2997	3031	3040	3030	3035	3035
C-15 N60 26 3969 3875 3978 3979 3969 3979 3979 35 3496 3492 3499 3496 3496 3500 3500 3500 3500 3500 3473 44 3475 3328 3481 3479 3472 3475 3473			15	3383	3383	3383	3383	3383	3383	3383
353496349234993496349635003500443475332834813479347234753473	C-15	N60	26	3969	3875	3978	3979	3969	3969	3979
44 3475 3328 3481 3479 3472 3475 3473			35	3496	3492	3499	3496	3496	3500	3500
			44	3475	3328	3481	3479	3472	3475	3473



Figure 2. The workflow of NM-ABC.

NM method is used in the onlooker bee phase of the ABC algorithm. The intention is to use the NM method in the onlooker bee phase instead of the employee bee phase since the individuals participating in the onlooker bee phase are selected based on probability. When the individual is chosen with probability, it is considered a quality individual in terms of fitness. Intensive search on quality individuals results in global optimum rather than searching on random individuals.

Algo	orithm 3: NM-ABC
1:	Initialize the population
2:	For $i = 1, 2,, FS$, do
3:	For $j = 1, 2,, S$, do
4:	Generate $x_{i,i}$ solution
5:	$x_{i,i} = x_{min,i} \pm rand(0,1) * (x_{max,i} - x_{min,i})$
6:	Where $x_{min,j}$ and $x_{max,j}$ are the min and max limit of the dimension j
7:	End for
8:	Compute the objective of the population
9:	iter = 1
10:	Repeat
11:	{
12:	Employed Bee Phase
13:	<i>For</i> each food source <i>i do</i>
14:	Generate candidate solution v_i using Equation (14)
15:	Select between v_i and x_i
16:	End For
17:	Onlooker Bee Phase
18:	Set $r = 0$
19:	While $(r \le FS)$
20:	If rand(0,1) < P_i using Algorithm 3
21:	Generate candidate solution v_i by Algorithm 2
22:	Select between v_i and x_i
23:	r = r + 1
24:	End if
25:	
26:	Scout bee Phase
27:	Abandon the food source x_i , which cannot improve further using Equation (13)
20:	iter – iter + 1
29: 30:	1001 = 1001 + 1
30. 31.	f Fud for
32.	Lill jui Ilutil iter – max FFs
32.	until hel – max res

4. Experimental Results

4.1. Experimental Setup

The NSP datasets are taken from the library NSP lib and consist of 25 to 100 instances; each contains "N" nurses, "D" days, and "S" shift patterns. The nurses' N varies from 25 to 100 nurses, and the schedule is to be made for given D days with S shift patterns. NSP lib contains cases to describe the maximum and minimum utilization of resources for the health care unit. It consists of "N" nurses, "D" days, and "S" shift patterns and provides the coverage matrix for days with shift patterns. The nurses' workload for the shift concerning a particular day relates to N and D for handling 25 to 100 instances.

The performance of NM-ABC for NSP is evaluated using the NSPLib dataset [42]. The dataset is accessed on 10 October 2021 from the specified link (https://www.projectmanagement. ugent.be/research/personnel_scheduling/nsp). The author summarizes the characteristic of the dataset in Table 1. The proposed technique to solve NSP is implemented with the help of MATLAB 2016a under Windows on an Intel i5 processor with 8 GB of RAM and 1 TB storage. Table 1 designates the cases utilized by NM-ABC to solve NSP.

The parameter settings of NM-ABC and compared algorithms are presented in Table 2. In this experimentation, algorithms M1 to M5 and NM-ABC consist of a population size of 100, and the maximum number of iterations is 1000. These algorithms will stop its execution when the maximum iterations or optimal solution are reached. The association of

the results obtained by NM-ABC clearly specifies that the proposed technique is relatively better than the existing methods.

4.2. Performance Metrics

The performance of the proposed technique is evaluated by relating it to five different opponent methods. Here, eight performance metrics are utilized to evaluate the experimental results listed in this section.

4.2.1. Average Best Time (ABT)

The best time is to achieve the best value for a particular instance. The average best time is the type of the best times of all test cases taken from the dataset. In experimental analysis, the average best time (ABT) is computed using

$$ABT = \sum_{i=1}^{n} \frac{\text{Time is taken to achieve the best value}}{\text{Total number of instances}}$$
(19)

where n is the number of test cases in the given data set.

4.2.2. Standard Deviation

Standard deviation (SD) is the portion of distribution among set values from its mean value. Average standard deviation determines the type of the standard deviation of all test cases taken from the dataset. The average standard deviation (ASD) can be measured using

$$ASD = \sqrt{\sum_{i=1}^{n} (value obtained in each instance_i - Mean value of the instance)^2}$$
(20)

where n is the number of instances in the given data set.

4.2.3. Least Error Rate

The least error rate (LER) is the variance between the actual optimal value and the obtained best value. The LER can be calculated using

$$LER = Value_{NSPLib} - Best \ Value_i \tag{21}$$

4.2.4. Success Percentage

Success percentage is the number of instances that attain optimal value for the given instance. Average success percentage (ASP) is the average number of models that achieve optimal value to the total number of test cases taken from the dataset. The value of ASP can be computed using

$$ASP = \sum_{i=1}^{n} \frac{\text{Number of instances succeed to attain optimal value}}{\text{Total number of instances}} * 100$$
(22)

where n is the number of instances in the given data set.

4.2.5. Cost Reduction

Cost reduction is the variation between actual cost in NSPLib and the cost obtained from our approach. Average cost reduction (ACR) is the average cost reduction to the total number of test cases taken from the dataset. The value of ACR can be computed from

$$ACR = \sum_{i=1}^{n} \frac{Cost_i - Cost_{NSPLib}}{\text{Total number of instances}}$$
(23)

where n is the number of test cases in the given data set.

4.2.6. Gap

The value of gap is the distance to attain the best deal. The average gap (AGap) is the average distance to achieve the best value from all instances of the total number of cases taken from the dataset. The value of AGap is calculated using

$$ACR = \sum_{i=1}^{n} \frac{(Cost_i - Cost_{NSPLib}) / Cost_{NSPLib}}{\text{Total number of instances}} * 100$$
(24)

where n is the number of instances in the given data set.

4.2.7. #Both Feasible Solution

#Both feasible solution (#BFS) is the number of feasible solutions found to obtain optimal value concerning both NSPLib and our approach's best value.

4.2.8. #Feasible Solution

#Feasible solution (#FS) is the number of feasible solutions found to obtain optimal value concerning known optimal values of NSPLib.

4.3. Experimental Result Analysis

The results obtained by NM-ABC with competitive methods are shown in Table 3. The performance is associated with exiting techniques; the value in the table defines the attained best value by proposed and other competitor's algorithms. The objective of NSP is to reduce cost; the lowermost principles are the best solution obtained. In the evaluation of the performance of the algorithm, the authors have considered 15 different cases of diverse sizes with five other instances. It is proven proposed NM-ABC accomplished 44 best results out of 75 instances.

The experimental results of 15 cases are summarized, and the best results are achieved within the time. The best solutions for each case in all methods are highlighted in bold font. The best values are obtained by using our proposed Algorithm 1. It is noted from Table 4 that, experimentally, NM-ABC obtained 12 best results out of 25 instances in smaller-sized datasets from case1 to case5, 14 best results in medium-sized datasets from case6 to case10, and 17 best results out of 25 instances in larger-sized datasets case12 to case15. The experimental results of the proposed algorithm have a high potential in exploiting search space solutions towards better results in various ways.

Table 4 shows the best time, standard deviation, and the least error rate for each case recorded for ten runs. The mean value of the proposed algorithm is 1.75% reduced compared to that of other competitive methods, showing that our proposed algorithm attained a lesser worst value in addition to the best solution. The least error rate for the proposed algorithm can be calculated using Equation (21) with the known optimal value recorded in NSPLib. The standard deviation is increased by 10% compared to other competitive methods. The computational time to attain each best value is shown, and the time taken to reach the best solution for each case is tabulated under the best time in a table. Our proposed method yields 39.32% less computational time to attain the best results than other competitor methods.

Table 5 describes the number of feasible solutions obtained by NM-ABC and other methods, and the table shows our proposed algorithm produced a more feasible solution than the solutions recorded in NSPLib. The solution is viable if the hard and soft constraints listed in Tables 4 and 5 are satisfied. NM-ABC satisfied several feasible solutions. #Both feasible is the number of possible solutions recorded in NSPLib or NM-ABC algorithm, and #feasible is the number of viable solutions obtained in NSPLib. Table 5 shows NM-ABC has attained 90% of a better value than known values reported in NSPLib. Compared with other methods, our algorithm outperforms with 87% of the best value, which is higher than other methods listed in Table 5. Thus, our algorithm contributes a new deterministic search and practical heuristic approach to solve NSP using NSPLib dataset.

				NM-A	BC		M ₁			M ₂			M ₃			M4			M ₅	
Case	Туре	Instance	SD	LER	Best Time	SD	LER	Best Time	SD	LER	Best Time	SD	LER	Best Time	SD	LER	Best Time	SD	LER	Best Time
C-1	N25	1	1.36	0	13.14	1.28	0	17.71	1.17	0	23.14	1.17	0	20.64	1.25	1	32	1.18	0	24.87
C-1	N25	7	2.40	4	11.09	1.27	1	13.82	1.45	1	21.09	1.45	1	26.09	1.85	1	19.14	1.37	1	19.91
C-1	N25	12	1.9	0	0.01	0.00	1	10.05	0.00	0	10.01	0.00	0	20.51	0.00	0	0.27	0.00	0	2.11
C-1	N25	19	1.80	0	0.5	1.47	0	10.71	1.96	0	0.5	1.96	1	12	1.79	0	1.5	1.85	0	3.70
C-1	N25	25	1.90	1	0.9	1.96	1	11.01	2.44	2	30.9	2.44	1	11.05	1.47	0	0.76	1.73	0	2.83
C-2	N25	2	3.70	5	19.36	2.36	3	21.94	2.32	5	22.5	2.32	2	23.18	2.04	1	24.09	1.91	1	5.28
C-2	N25	5	1.76	0	1.54	1.96	0	12.33	2.15	1	23.44	2.24	1	34.21	2.16	1	35.55	2.86	2	6.50
C-2	N25	9	1.85	0	0.05	1.99	0	12.19	1.87	0	30	1.87	0	20.03	1.95	0	15.01	1.81	1	13.63
C-2	N25	15	4.43	7	21.2	2.71	3	21.71	2.49	4	29.2	2.34	3	12.8	2.10	0	23.6	2.14	4	4.84
C-2	N25	27	1.55	0	0.04	1.92	0	0.36	1.57	0	25	1.57	0	31.52	1.96	0	32.26	2.12	0	3.42
C-3	N25	1	2.52	2	1.2	1.70	0	1.67	1.91	6	12	2.14	4	22.6	2.14	1	23.3	1.89	1	4.60
C-3	N25	3	1.60	0	0.34	1.62	0	0.43	1.87	0	10.34	1.87	0	10.51	1.45	2	10.6	2.10	0	2.18
C-3	N25	16	1.79	2	23.07	1.91	1	30.32	2.06	0	28	1.91	3	21.54	1.62	3	50.77	1.92	0	49.14
C-3	N25	27	2.84	2	15.73	1.64	0	20.58	1.99	0	20	2.10	1	27.87	1.89	0	33.93	2.50	1	33.47
C-3	N25	35	4.35	6	3.28	2.69	3	4.32	2.95	3	34.3	2.79	2	25.94	1.73	0	7.27	1.56	0	8.40
C-4	N25	5	9.48	28	32.13	4.86	12	40.56	5.42	6	34	4.32	3	50.07	2.33	1	59.03	2.94	3	58.47
C-4	N25	10	5.61	11	12.67	4.57	9	19.34	4.27	4	29	3.82	7	35.34	2.24	3	46.67	3.53	5	42.74
C-4	N25	25	5.82	14	22.5	3.01	5	30.5	3.52	8	33.6	3.35	3	44.85	1.96	2	56.03	2.69	4	53.17
C-4	N25	38	1.74	0	17.29	1.95	0	22.05	1.95	0	19.34	1.95	0	27.99	2.07	1	33.33	2.44	5	33.45
C-4	N25	41	1.89	0	1.54	1.67	0	2.38	1.66	0	3.67	1.66	0	4.44	1.66	3	5.89	1.79	0	6.77
C-5	N25	7	3.12	4	17.32	2.45	6	10.64	2.50	6	14.26	2.27	0	17.92	1.83	0	23.22	1.85	3	22.37
C-5	N25	11	1.64	0	0.07	2.11	0	9.69	1.90	0	2.89	1.90	0	2.93	2.19	1	4.35	2.15	0	5.10
C-5	N25	30	2.30	2	15.3	2.24	1	23.64	2.24	8	4.3	2.24	4	6.95	1.72	0	2.36	1.83	1	9.41
C-5	N25	42	1.97	0	3.29	1.95	4	8.32	2.18	0	4.26	2.18	0	5.91	2.24	0	7.21	1.85	0	8.36
C-5	N25	47	2.34	1	14.2	2.20	1	18.28	2.62	5	65	2.62	1	67.1	2.33	1	98.55	2.02	1	81.70
C-6	N50	1	6.14	6	29.4	6.46	4	15.95	4.82	2	29	4.65	2	33.7	4.58	0	45.85	5.23	2	41.10
C-6	N50	4	4.49	1	16.1	4.68	0	9.31	3.88	1	33.97	3.88	0	27.02	5.17	1	22.48	3.69	8	21.41
C-6	N50	12	5.06	4	18.45	4.85	3	10.92	3.99	5	25.3	3.56	0	29.53	4.24	0	17.29	5.29	0	24.19
C-6	N50	26	5.06	0	1.56	4.08	3	85.62	4.26	0	23.92	4.26	0	32.78	3.72	0	18.39	3.83	1	17.78
C-6	N50	29	3.77	0	1.43	4.76	3	9.01	3.31	0	31.78	3.31	0	22.5	5.06	2	53.03	4.21	2	4.45
C-7	N50	3	30.48	78	34.67	4.35	0	44.55	3.10	7	40.32	3.10	5	57.66	3.74	11	69.15	4.74	9	67.32
C-7	N50	6	4.10	0	3.33	4.07	0	9.35	3.26	0	34.2	3.26	0	35.87	3.87	0	37.13	3.52	0	8.31
C-7	N50	12	5.76	6	59.02	5.15	3	72.39	4.58	3	24.2	4.45	2	18.71	3.52	0	23.56	5.82	6	23.15

Table 4. Summary of standard deviation, least error, and best time obtained by proposed Artificial bee colony with Nelder Mead method (NM-ABC) and competitor methods.

Table 4. Cont.

				NM-A	BC		M ₁			M ₂			M ₃			M ₄			M_5		
Case	Туре	Instance	SD	LER	Best Time	SD	LER	Best Time	SD	LER	Best Time	SD	LER	Best Time	SD	LER	Best Time	SD	LER	Best Time	
C-7	N50	26	5.44	5	18.45	4.47	1	30.48	4.84	1	38.05	4.84	0	12.28	4.27	1	14.19	4.38	2	15.49	
C-7	N50	36	4.88	0	1.36	4.37	0	11.84	4.45	0	22	4.45	0	42.68	4.15	0	33.34	4.73	0	4.68	
C-8	N50	4	5.16	4	33.9	4.75	2	44.66	5.28	1	12.9	5.28	2	14.85	5.36	4	45.33	4.96	1	7.03	
C-8	N50	9	3.96	0	21.28	4.67	0	32.31	3.88	0	34.6	3.88	0	15.24	4.22	0	37.22	4.58	0	7.76	
C-8	N50	15	7.28	7	32.4	4.87	3	35.82	4.56	9	23.9	4.04	0	20.1	4.36	3	23.95	3.71	5	24.48	
C-8	N50	40	5.07	0	12.25	4.87	3	23.92	4.47	0	27.4	4.47	0	28.53	4.47	0	61.66	5.06	1	11.61	
C-8	N50	47	4.27	0	20.65	4.05	10	25.33	4.96	0	37.23	5.08	1	89.06	4.04	3	41.76	4.64	0	12.10	
C-9	N60	5	28.31	63	32.43	8.94	8	43.29	7.79	0	45.29	7.01	0	61.51	5.75	6	76.04	5.76	10	72.16	
C-9	N60	10	9.70	7	12.34	6.14	2	16.29	8.53	5	26.36	8.63	3	22.53	6.12	0	27.63	7.25	9	27.40	
C-9	N60	23	16.40	26	23.54	7.26	3	31.13	5.80	4	31.5	5.69	1	43.27	7.34	1	53.14	5.54	1	51.17	
C-9	N60	29	22.01	36	43.67	8.98	8	53.8	7.19	0	39.98	6.04	8	61.82	6.44	0	70.89	6.53	1	71.41	
C-9	N60	40	5.47	0	0.37	5.57	0	10.68	5.15	0	21.4	5.15	0	11.59	6.15	1	12.19	6.41	0	3.47	
C-10	N60	6	6.09	0	20.11	5.75	0	23.49	5.51	0	32.4	5.51	0	42.46	5.73	0	23.63	6.33	0	4.54	
C-10	N60	14	7.71	4	42.03	5.08	1	43.13	6.50	5	24.78	6.50	5	35.8	6.46	0	47.68	7.28	6	8.35	
C-10	N60	20	6.40	0	24.65	6.62	0	30.24	5.64	0	42	5.64	0	54.33	6.05	1	39.16	6.89	0	18.33	
C-10	N60	32	7.49	0	20.04	6.89	1	31.31	6.96	2	65.9	6.83	0	25.92	6.33	0	48.86	6.87	3	8.70	
C-10	N60	41	9.67	10	55.09	8.22	4	79.36	7.54	6	37.43	7.17	1	44.98	5.77	0	59.92	5.55	6	30.06	
C-11	N60	2	6.96	0	35.07	6.24	0	46.6	6.85	0	56.3	6.85	0	58.84	7.07	0	30.72	6.08	0	11.70	
C-11	N60	8	5.34	0	27.3	6.10	0	39.55	6.28	0	69.26	6.28	0	52.91	6.17	0	55.72	6.60	0	16.36	
C-11	N60	14	8.05	3	49.76	6.29	2	51.94	5.70	3	48.53	6.18	5	53.41	7.17	8	75.24	6.95	2	16.72	
C-11	N60	20	6.17	0	26.2	6.56	0	39.19	6.98	0	32.94	6.98	3	46.04	5.99	0	90.96	7.15	0	20.23	
C-11	N60	32	7.03	3	51.87	8.60	3	65.44	7.83	1	70.43	7.69	3	76.37	6.00	10	74.99	7.79	1	32.05	
C-12	N60	3	5.66	1	20.23	6.26	9	32.29	6.28	0	32.9	6.28	4	63.02	7.16	8	44.41	6.91	4	5.20	
C-12	N60	12	12.67	12	49.22	5.88	0	52.38	6.20	0	43.2	6.20	2	37.81	6.55	10	52.11	6.93	4	22.05	
C-12	N60	19	8.26	0	21.28	6.87	0	42.37	8.26	1	54.89	8.26	0	35.53	6.90	10	57.66	5.59	0	8.11	
C-12	N60	23	8.72	5	19.9	7.42	1	34.68	6.53	0	25.82	6.37	0	51.27	7.02	0	56.46	7.16	1	26.03	
C-12	N60	34	6.58	0	39	7.54	1	49.39	5.68	0	72	5.50	0	61.5	7.19	0	72.75	7.34	0	71.50	
C-13	N60	1	7.73	1	33.11	6.36	0	44.33	6.13	0	45.19	6.13	0	56.75	6.08	5	58.56	6.96	5	9.38	
C-13	N60	4	12.77	19	47.93	9.10	1	52.95	7.89	3	50.45	7.61	8	49.42	7.12	1	35.16	7.11	13	35.11	
C-13	N60	19	9.93	11	44.43	9.06	5	49.08	7.30	1	39.29	6.00	3	56.51	6.00	0	62.54	8.27	3	31.96	
C-13	N60	29	8.06	3	35.9	7.35	2	38.85	7.09	0	61.45	6.78	4	64.4	7.44	1	40.65	6.58	0	18.29	
C-13	N60	40	6.54	0	32.71	7.26	0	44.96	5.95	0	65.23	5.95	0	57.09	7.49	0	78.77	5.73	0	9.73	
C-14	N60	5	13.00	18	18.43	7.87	2	23.5	6.45	6	43.9	6.80	6	53.12	8.35	7	35.46	7.08	6	39.50	
C-14	N60	9	11.52	14	10.05	7.49	0	20.6	7.43	0	52.56	7.43	10	42.59	7.15	10	23.85	6.85	0	4.70	

Tab	le 4.	Cont.	

				NM-Al	BC	M ₁				M ₂		M ₃				M_4		M_5			
Case	Туре	Instance	SD	LER	Best Time	SD	LER	Best Time	SD	LER	Best Time	SD	LER	Best Time	SD	LER	Best Time	SD	LER	Best Time	
C-14	N60	15	28.15	91	49.02	7.63	0	60.4	6.61	7	53	7.44	5	77.51	7.57	6	79.63	7.21	3	89.71	
C-14	N60	30	22.13	62	47.34	10.30	7	60.3	6.05	1	52.6	5.08	1	76.27	6.05	1	90.74	6.38	1	88.39	
C-14	N60	43	6.02	2	37.1	6.56	1	34	6.90	0	49	6.73	0	28.55	5.76	0	34.28	5.22	0	34.15	
C-15	N60	6	17.40	34	20.47	13.08	0	25.47	10.86	9	59.9	7.59	1	30.14	5.71	4	34.97	7.24	4	35.72	
C-15	N60	15	5.85	0	24.23	6.17	0	35.86	5.09	0	36.9	5.09	0	39.02	5.81	0	51.41	6.51	0	12.00	
C-15	N60	26	43.30	94	79.38	12.15	9	100.5	9.61	10	85.34	6.57	0	125.03	5.18	0	147.86	6.73	10	143.70	
C-15	N60	35	7.13	4	51.28	7.05	3	64.02	6.14	0	59.25	6.14	0	79.89	4.77	4	89.2	6.18	4	13.34	
C-15	N60	44	57.34	147	84.3	6.45	6	96.95	7.47	4	87	7.89	3	87.15	6.81	0	88.58	6.16	2	97.75	

Case	Туре	Instances	NM-ABC		M_1		M_2		M_3		\mathbf{M}_4		Μ	l ₅
			#BFS	#FS	#BFS	#FS	#BFS	#FS	#BFS	#FS	#BFS	#FS	#BFS	#FS
1	N25	1, 7, 12, 19, 25	3	5	2	4	3	4	2	4	2	4	4	4
2	N25	2, 5, 9, 15, 27	3	5	3	3	2	3	2	3	3	5	1	4
3	N25	1, 3, 16, 27, 35	1	4	3	5	3	4	1	2	1	3	3	5
4	N25	5, 10, 25, 38, 41	2	5	2	5	2	3	2	5	0	4	1	3
5	N25	7, 11, 30, 42, 47	2	5	1	3	2	2	3	4	3	5	2	4
6	N50	1, 4, 12, 26, 29	2	5	0	2	2	3	4	5	3	5	1	2
7	N50	3, 6,12, 26, 36	2	5	3	4	2	4	2	2	3	3	2	2
8	N50	4, 9, 15, 40, 47	3	5	1	3	3	3	3	5	2	2	2	3
9	N50	5, 10, 23, 29, 40	1	5	1	2	3	3	2	3	2	4	1	2
10	N50	6,14, 20, 32, 41	3	5	3	5	2	2	3	4	4	5	2	2
11	N60	2, 8, 14, 20, 32	3	5	3	4	3	4	2	2	3	3	3	4
12	N60	3, 12, 19, 23, 34	2	5	2	4	4	5	3	3	2	2	2	3
13	N60	1, 4, 19, 29, 40	1	5	2	2	3	5	2	2	2	4	2	2
14	N60	5, 9, 15, 30, 43	0	4	2	5	2	4	1	3	1	2	2	4
15	N60	6, 15, 26, 35, 44	1	5	2	2	4	3	2	4	2	3	2	2

Table 5. Comparison of the number of #both feasible and #feasible solutions obtained by the proposedArtificial bee colony with Nelder Mead method (NM-ABC) and competitor's methods.

Table 6 summarizes the comparison and assessment of average best time, average standard deviation, and average success percentage obtained by our proposed algorithm NM-ABC with another competitor method. In Table 4, the columns 5 and 6 describe the average best time, average standard deviation, and average success percentage of the proposed algorithm. Columns 7 to 21 describe the performance metrics of another competitor's method.

Table 6. Comparison and assessment of ABT, ASD, and ASP obtained by proposed algorithm NM-ABC and competitor methods.

Case	Trues	Instances	NM-ABC			M ₁ M			M ₂	I ₂ M ₃					M 4			M 5		
	Type		ABT	ASD	ASP	ABT	ASD	ASP	ABT	ASD	ASP	ABT	ASD	ASP	ABT	ASD	ASP	ABT	ASD	ASP
1	N25	1, 7, 12, 19, 25	5.13	1.87	100	12.66	1.2	80	17.13	1.4	80	18.06	1.4	80	10.73	1.27	80	18.68	1.23	80
2	N25	2, 5, 9, 15, 27	8.44	2.66	100	13.71	2.19	60	26.03	2.08	60	24.35	2.07	60	26.1	2.04	100	24.73	2.17	80
3	N25	1, 3, 16, 27, 35	8.72	2.62	80	11.46	1.91	100	20.93	2.16	80	21.69	2.16	40	25.17	1.77	60	31.56	1.99	100
4	N25	5, 10, 25, 38, 41	17.23	4.91	100	22.97	3.21	100	23.92	3.36	60	32.54	3.02	100	40.19	2.05	80	38.92	2.68	60
5	N25	7, 11, 30, 42, 47	10.04	2.27	100	14.11	2.19	60	18.14	2.29	40	20.16	2.24	80	27.14	2.06	100	31.39	1.94	80
6	N50	1, 4, 12, 26, 29	13.39	4.9	100	26.16	4.97	40	28.79	4.05	60	29.11	3.93	100	31.41	4.55	100	33.19	4.45	40
7	N50	3, 6,12, 26, 36	23.37	10.13	100	33.72	4.48	80	31.75	4.05	80	33.44	4.02	40	35.47	3.91	60	37.79	4.64	40
8	N50	4, 9, 15, 40, 47	24.1	5.15	100	32.41	4.64	60	27.21	4.63	60	33.56	4.55	100	41.98	4.49	40	44.6	4.59	60
9	N50	5, 10, 23, 29, 40	22.47	16.38	100	31.04	7.38	40	32.91	6.89	60	40.14	6.5	60	47.98	6.36	80	45.12	6.3	40
10	N50	6,14, 20, 32, 41	32.38	7.47	100	41.51	6.51	100	40.5	6.43	40	40.7	6.33	80	43.85	6.07	100	45.2	6.58	40
11	N60	2, 8, 14, 20, 32	38.04	6.71	100	48.54	6.76	80	55.49	6.73	80	57.51	6.8	40	65.53	6.48	60	59.41	6.91	80
12	N60	3, 12, 19, 23, 34	29.93	8.38	100	42.22	6.79	80	45.76	6.59	100	49.83	6.52	60	56.68	6.96	40	60.58	6.79	60
13	N60	1, 4, 19, 29, 40	38.82	9.01	100	46.03	7.83	40	52.32	6.87	100	56.83	6.49	40	55.14	6.83	80	58.89	6.93	40
14	N60	5, 9, 15, 30, 43	32.39	16.16	80	39.76	7.97	100	50.21	6.69	80	55.61	6.7	60	52.79	6.98	40	61.29	6.55	80
15	N60	6, 15, 26, 35, 44	51.93	26.2	100	64.56	8.98	40	65.68	7.83	60	72.25	6.66	80	82.4	5.66	60	99.3	6.56	40

Figure 3 compares the average best time of NM-ABC with other methods. The best time is the time taken to attain the best value, for instance, using Algorithm 2. The time is taken to allocate and schedule nurses for a particular time without overruling hard constraints and reducing violations of soft constraints. For smaller datasets, the computational time taken by NM-ABC is reduced to 56.72%. For medium datasets, the time taken is reduced by 36.40%. For larger datasets, the time taken is reduced by 34.31% compared to other competitor methods. The ABT is calculated using Equation (19) and shows the reduced computational time to schedule nurses for a particular shift on days for the scheduled period. The HSHH algorithm consumes more computational time to solve NSP, while MAPA and BCO achieve 50% of our proposed approach.



Figure 3. Analysis and Comparison of Average Best time.

Figure 4 portrays the comparison of the average standard deviation of NM-ABC with another competitor method. Figure 4 shows that an increase in the standard deviation will increase the search space to obtain the best value.



Figure 4. Analysis and Comparison of Average Standard Deviation.

Figure 5 compares our proposed algorithm NM-ABC algorithm's average success percentage with other methods. Success percentage is the number of instances in attaining optimal value for the given instance. Average success percentage (ASP) is the average number of instances that obtains optimal value for the total number of cases taken from the dataset. ASP is calculated using Equation (22) and proves NM-ABC has increased the

success percentage over other competing methods. In Table 7, it is observed that NM-ABC achieved a 100% success percentage except for case 3 and case 14. Our algorithm NM-ABC shows an overall 97.33% of success percentage. For smaller datasets, it is shown that the success percentage is 96% and 26% more than that of all other competitor methods. For the medium dataset, our algorithm achieved a 100% success percentage of 56.25% more than other competing methods. The larger dataset NM-ABC attained a 96% success percentage, which is 48.18% more than other methods. In the competitor method, the multi-objective ant colony optimization algorithm (M4) achieved the second-best success percentage with an overall 72%.



Figure 5. Analysis and Comparison of Average Success Percentage.

 Table 7. Comparison and assessment of ACR and AGap obtained by proposed algorithm NM-ABC and competitor methods.

Case	Туре	Testeres	NM-ABC		\mathbf{M}_{1}		M ₂		M ₃		M_4		Μ	5
		Instances	ACR	AGap	ACR	AGap	ACR	AGap	ACR	AGap	ACR	AGap	ACR	AGap
1	N25	1, 7, 12, 19, 25	1.00	0.33	0.20	0.07	0.20	0.07	0.20	0.07	0.00	0.00	-0.20	-0.07
2	N25	2, 5, 9, 15, 27	2.40	0.83	-1.20	-0.42	-1.60	-0.55	-0.80	-0.28	0.40	0.14	0.00	0.00
3	N25	1, 3, 16, 27, 35	1.60	0.49	0.80	0.25	-0.60	-0.18	-1.20	-0.37	-0.80	-0.25	0.40	0.12
4	N25	5, 10, 25, 38, 41	10.60	3.55	5.20	1.74	-0.40	-0.13	2.60	0.87	0.80	0.27	-1.40	-0.47
5	N25	7, 11, 30, 42, 47	1.40	0.47	-1.60	-0.53	-3.80	-1.27	-0.60	-0.20	0.40	0.13	-0.20	-0.07
6	N50	1, 4, 12, 26, 29	2.20	0.38	-1.40	-0.24	-1.20	-0.20	0.40	0.07	0.60	0.10	-2.20	-0.38
7	N50	3, 6,12, 26, 36	17.80	2.99	-0.40	-0.07	-0.60	-0.10	-1.40	-0.24	-2.40	-0.40	-3.40	-0.57
8	N50	4, 9, 15, 40, 47	2.20	0.38	-1.60	-0.27	-2.00	-0.34	0.60	0.10	-2.00	-0.34	-1.00	-0.17
9	N50	5, 10, 23, 29, 40	26.40	0.84	-3.40	-0.11	-1.80	-0.06	-2.00	-0.06	-0.80	-0.03	-3.80	-0.12
10	N50	6,14, 20, 32, 41	2.80	0.08	1.20	0.04	-2.60	-0.08	-0.80	-0.02	0.20	0.01	-3.00	-0.09
11	N60	2, 8, 14, 20, 32	1.20	0.03	0.20	0.01	-0.40	-0.01	-2.20	-0.06	-3.60	-0.10	-0.20	-0.01
12	N60	3, 12, 19, 23, 34	3.60	0.12	-1.40	-0.05	0.20	0.01	-1.20	-0.04	-5.60	-0.18	-1.40	-0.05
13	N60	1, 4, 19, 29, 40	6.80	0.22	-1.60	-0.05	0.80	0.03	-3.00	-0.10	-0.60	-0.02	-4.20	-0.14
14	N60	5, 9, 15, 30, 43	36.60	1.20	2.00	0.07	0.40	0.01	-2.00	-0.07	-4.40	-0.14	-0.40	-0.01
15	N60	6, 15, 26, 35, 44	55.80	1.61	-3.60	-0.10	-4.60	-0.13	0.80	0.02	-1.60	-0.05	-3.20	-0.09

Table 7 summarizes the comparison and assessment of average cost reduction and average gap percentage obtained by NM-ABC with another competitor method. Table 7,

the 4th and 5th columns describe the average cost reduction and average gap percentage of the proposed algorithm. Columns 6 to 15 illustrate the performance metrics of another competitor's method.

Figure 6 portrays the analysis and comparison of the average cost reduction of our proposed NM-ABC with another competitor method. ACR is the difference between the best-known value observed in NSPLib and the cost obtained from our algorithm. Average cost reduction (ACR) is the average cost reduction from the dataset to the total number of instances and calculated using Equation (23). In Table 7, it is shown that NM-ABC minimized the cost of the NSP. The main objective of NSP is to reduce resource utilization, which is reflected by the cost reduction, as shown in Figure 6. Using our proposed algorithm, NM-ABC, the cost of NSP is reduced by 0.66%. For more minor instances, NM-ABC is reduced to 1.12% compared to other competing methods and is reduced to 0.11%. For medium instances, our proposed algorithm is reduced to 0.62% more than the original cost value recorded in NSPLib, and other methods reduced it to 0.70%. Compared to more significant instances, the proposed algorithm is reduced to 0.63%; compared with other competitor methods, it is reduced to 0.68%. In the proposed NM-ABC algorithm, the cost of resource utilization decreases with an increase in the dataset size.



Figure 6. Analysis and Comparison of Average Cost Reduction.

Figure 7 compares the average gap percentage of NM-ABC with another competitor method. The gap percentage is the distance between the attained best value and the known optimal value recorded in NSPLib. The average gap (AGap) is the average distance to obtain the best and known value from all instances to the total number of instances. The value of AGap is calculated using Equation (23), and from Table 7, it is proven that NM-ABC attained a positive value, which shows the algorithm moved towards the best optimum value. Our algorithm achieved 94.30% of successfully solved instances to reach the best value concerning the known value recorded in NSPLib.



Figure 7. Analysis and Comparison of Average Gap.

Cases

5. Discussions

The experiments to solve NP-hard combinatorial NSP are conducted by our proposed algorithm NM-ABC. Problem-based existing algorithms are chosen and compared with the proposed NM-ABC algorithm. The result of our proposed algorithm is compared with other competitor methods, and the best values are compared in Table 7. Various performance metrics are considered to evaluate the proposed algorithm's performance. Tables 3–7 show the outcome of our proposed algorithm and other existing method's performance. From the table and figure, it is evident that NM-ABC has more ability to attain the best value with less computation time compared to the known optimal value listed in NSPLib. The average number of function evaluations (NFEs) for proposed NM-ABC is observed concerning to number solutions updated using reflection, contraction, expansion, and shrinkage phase. We noticed that NFEs of proposed algorithm is 10⁶ for all the test cases.

Compared with other existing methods, the mean value of NM-ABC is reduced by 1.75% compared to that of other competitive methods and attained a lesser worst value in addition to the best solution. The proposed method yields 39.32% less computational time to obtain best results compared to other competitor methods. The datasets are divided based on their size as smaller, medium, and large datasets; the computational time taken by NM-ABC is reduced to 56.72%, 36.40%, and 34.31%, respectively. The success percentage to attain the best value of our proposed approach is 97.33%. Compared with other methods with various-sized datasets, our algorithm achieves 26% for the smaller dataset, 56.25% for medium datasets, and 48.18% for larger datasets. The cost of our algorithm is reduced by 0.66%, and the gap percentage to move towards the optimum value is that 94.30% instances were successfully solved to obtain the best deal with the known optimal value recorded in NSPLib.

Our algorithm has proven significant performance in attaining the best solution with optimized resource utilization and nurse preferences by satisfying both hard and soft constraints. It is also shown that the existing approach solves NSP with higher utilization of resources and violation of soft constraints that lead to increased cost. The ability to distribute the workload among nurses with nurse performance and satisfaction are achieved in our algorithm. The proposed system is tested on larger datasets and works astoundingly well than the other techniques.

6. Conclusions

This paper solves NSP using a hybrid artificial bee colony algorithm with Nelder-Mead (NM-ABC) method. The proposed algorithm is evaluated using the NSPLib dataset, and the performance of the proposed algorithm is compared with the other five existing methods and evaluated in the NSPLib dataset. To assess the implementation of our proposed algorithm, 75 different cases of various-sized datasets were chosen for evaluation, and in that, 44 out of 75 instances achieved the best result. The evaluation of the proposed algorithm is compared with other existing techniques in terms of the best time, standard deviation, least error rate, success percentage, cost reduction, average gap, #both feasible solutions, and the number of #feasible solutions. When comparing the results of existing algorithms in metrics listed, the proposed NM-ABC outperforms in most instances of NSP. Future work of this research can be extended with more objectives in NSP for optimization.

Author Contributions: Conceptualization, R.M.; methodology, R.M. and R.R.; validation, S.S.A. and M.R.; formal analysis, A.G., R.S. and A.D.; writing—original draft preparation, R.M.; writing—review and editing, M.R. and R.S.; supervision, R.R. and D.G.; funding acquisition, S.S.A. All authors have read and agreed to the published version of the manuscript.

Funding: This study was funded by the Deanship of Scientific Research, Taif University Researchers Supporting Project number (TURSP-2020/215), Taif University, Taif, Saudi Arabia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data in this research paper will be shared upon request to the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Anwar, K.; Awadallah, M.A.; Khader, A.T.; Al-Betar, M.A. Hyper-heuristic approach for solving nurse rostering problem. In Proceedings of the 2014 IEEE Symposium on Computational Intelligence in Ensemble Learning (CIEL), Orlando, FL, USA, 9–12 December 2014.
- 2. Awadallah, M.A.; Bolaji, A.L.; Al-Betar, M.A. A hybrid artificial bee colony for a nurse rostering problem. *Appl. Soft Comput.* **2015**, 35, 726–739. [CrossRef]
- Constantino, A.A.; Landa-Silva, D.; de Melo, E.L.; de Mendonça, C.F.X.; Rizzato, D.B.; Romão, W. A heuristic algorithm based on multi-assignment procedures for nurse scheduling. *Ann. Oper. Res.* 2014, 218, 165–183. [CrossRef]
- 4. Garey, M.R.; Johnson, D.S. Computers and Intractability: A Guide to the Theory of NP-Completeness; W.H. Freeman: San Francisco, CA, USA, 1979.
- 5. Megeath, J.D. Successful hospital personnel scheduling. Interfaces 1978, 8, 55-60. [CrossRef]
- Musliu, N.; Gärtner, J.; Slany, W. Efficient generation of rotating workforce schedules. *Discret. Appl. Math.* 2002, 118, 85–98. [CrossRef]
- Millar, H.H.; Kiragu, M. Cyclic and non-cyclic scheduling of 12 h shift nurses by network programming. *Eur. J. Oper. Res.* 1998, 104, 582–592. [CrossRef]
- 8. Burke, E.; de Causmaecker, P.; Berghe, G.V. A hybrid tabu search algorithm for the nurse rostering problem. In *Simulated Evolution and Learning. SEAL 1998*; Springer: Berlin/Heidelberg, Germany, 1998; pp. 187–194. [CrossRef]
- 9. Legrain, A.; Omer, J.; Rosat, S. An online stochastic algorithm for a dynamic nurse scheduling problem. *Eur. J. Oper. Res.* 2020, 285, 196–210. [CrossRef]
- 10. Özcan, E.; Bilgin, B.; Korkmaz, E.E. Hill climbers and mutational heuristics in hyperheuristics. In *Parallel Problem Solving from Nature-PPSN IX*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 202–211.
- 11. Cheang, B.; Li, H.; Lim, A.; Rodrigues, B. Nurse rostering problems—A bibliographic survey. *Eur. J. Oper. Res.* 2003, 151, 447–460. [CrossRef]
- 12. Jaradat, G.M.; Al-Badareen, A.; Ayob, M.; Al-Smadi, M.; Al-Marashdeh, I.; Ash-Shuqran, M.; Al-Odat, E. Hybrid elitist-ant system for nurse-rostering problem. *J. King Saud Univ. Comput. Inf. Sci.* **2019**, *31*, 378–384. [CrossRef]
- 13. Van den Bergh, J.; Beliën, J.; De Bruecker, P.; Demeulemeester, E.; De Boeck, L. Personnel scheduling: A literature review. *Eur. J. Oper. Res.* **2013**, *226*, 367–385. [CrossRef]
- 14. Burke, E.K.; De Causmaecker, P.; Berghe, G.V.; Van Landeghem, H. The state of the art of nurse rostering. *J. Sched.* **2004**, *7*, 441–499. [CrossRef]

- 15. Aickelin, U.; Dowsland, K.A. Exploiting problem structure in a genetic algorithm approach to a nurse rostering problem. *J. Sched.* **2000**, *3*, 139–153. [CrossRef]
- Burke, E.K.; Li, J.; Qu, R. Pareto-Based Optimization for Multi-objective Nurse Scheduling. Computer Science Technical Report No. NOTTCS-TR-2007-5. 2007. Available online: https://www.researchgate.net/publication/277291801_Pareto-Based_Optimization_ for_Multi-objective_Nurse_Scheduling_Corresponding_author (accessed on 1 October 2021).
- 17. Weil, G.; Heus, K.; Francois, P.; Poujade, M. Constraint programming for nurse scheduling. *IEEE Eng. Med. Biol. Mag.* **1995**, 14, 417–422. [CrossRef]
- 18. Brusco, M.J.; Jacobs, L.W. Cost analysis of alternative formulations for personnel scheduling in continuously operating organizations. *Eur. J. Oper. Res.* **1995**, *86*, 249–261. [CrossRef]
- Lourenço, H.R.; Martin, O.C.; Stützle, T. Iterated local search. In *Handbook of Metaheuristics*; Springer: New York, NY, USA, 2003; pp. 320–353.
- 20. Warner, D.M.; Prawda, J. A mathematical programming model for scheduling nursing personnel in a hospital. *Manag. Sci.* **1972**, 19 *Pt* 1, 411–422. [CrossRef]
- Kawanaka, H.; Yamamoto, K.; Yoshikawa, T.; Shinogi, T.; Tsuruoka, S. Genetic algorithm with the constraints for nurse scheduling problem. In Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546), Seoul, Korea, 27–30 May 2001; Volume 2, pp. 1123–1130.
- 22. Burke, E.K.; Li, J.; Qu, R. A hybrid model of integer programming and variable neighborhood search for highly-constrained nurse rostering problems. *Eur. J. Oper. Res.* 2010, 203, 484–493. [CrossRef]
- 23. Kennedy, J. Particle swarm optimization. In Encyclopedia of Machine Learning; Springer: New York, NY, USA, 2011; pp. 760–766.
- Gutjahr, W.J.; Rauner, M.S. An ACO algorithm for a dynamic regional nurse-scheduling problem in Austria. *Comput. Oper. Res.* 2007, 34, 642–666. [CrossRef]
- 25. Berrada, I.; Ferland, J.A.; Michelon, P. A multi-objective approach to nurse scheduling with both hard and soft constraints. *Socio-Econ. Plan. Sci.* **1996**, *30*, 183–193. [CrossRef]
- 26. Burke, E.K.; Curtois, T.; Qu, R.; Berghe, G.V. A scatter search methodology for the nurse rostering problem. *J. Oper. Res. Soc.* 2010, 61, 1667–1679. [CrossRef]
- 27. Todorovic, N.; Petrovic, S. Bee colony optimization algorithm for nurse rostering. *IEEE Trans. Syst. Man Cybern. Syst.* 2013, 43, 467–473. [CrossRef]
- 28. Xu, Y.; Wang, X. An artificial bee colony algorithm for scheduling call centres with weekend-off fairness. *Appl. Soft Comput.* **2021**, 109, 107542. [CrossRef]
- 29. Yang, M.; Ni, Y.; Yang, L. A multi-objective consistent home healthcare routing and scheduling problem in an uncertain environment. *Comput. Ind. Eng.* 2021, 160, 107560. [CrossRef]
- Basturk, B.; Karaboga, D. An artificial bee colony (ABC) algorithm for numeric function optimization. *IEEE Swarm Intell. Symp.* 2006, *8*, 687–697.
- Erkoc, M.E.; Karaboga, N. A novel sparse reconstruction method based on multi-objective Artificial Bee Colony algorithm. *Signal Process.* 2021, 189, 108283. [CrossRef]
- 32. Karaboga, D.; Basturk, B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *J. Glob. Optim.* **2007**, *39*, 459–471. [CrossRef]
- 33. Kong, D.; Chang, T.; Dai, W.; Wang, Q.; Sun, H. An improved artificial bee colony algorithm based on elite group guidance and combined breadth-depth search strategy. *Inf. Sci.* **2018**, *442*, 54–71. [CrossRef]
- 34. Naidu, K.; Mokhlis, H.; Terzija, V. Performance investigation of ABC algorithm in multi-area power system with multiple interconnected generators. *Appl. Soft Comput.* **2017**, *57*, 436–451. [CrossRef]
- Ozturk, C.; Hancer, E.; Karaboga, D. Dynamic clustering with improved binary artificial bee colony algorithm. *Appl. Soft Comput.* 2015, 28, 69–80. [CrossRef]
- 36. Su, S.; Zhou, F.; Yu, H. An artificial bee colony algorithm with variable neighborhood search and tabu list for long-term carpooling problem with time window. *Appl. Soft Comput.* **2019**, *85*, 105814. [CrossRef]
- 37. Zhao, H.; Pei, Z.; Jiang, J.; Guan, R.; Wang, C.; Shi, X. A hybrid swarm intelligent method based on genetic algorithm and artificial bee colony. In *Advances in Swarm Intelligence: ICSI 2010*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 558–565. [CrossRef]
- 38. Wickert, T.I.; Smet, P.; Berghe, G.V. The nurse rerostering problem: Strategies for reconstructing disrupted schedules. *Comput. Oper. Res.* **2019**, 104, 319–337. [CrossRef]
- 39. Wolbeck, L.; Kliewer, N.; Marques, I. Fair shift change penalization scheme for nurse rescheduling problems. *Eur. J. Oper. Res.* **2020**, *284*, 1121–1135. [CrossRef]
- 40. Zhu, G.; Kwong, S. Gbest-guided artificial bee colony algorithm for numerical function optimization. *Appl. Math. Comput.* **2010**, 217, 3166–3173. [CrossRef]
- Črepinšek, M.; Liu, S.-H.; Mernik, M. Exploration and exploitation in evolutionary algorithms: A survey. ACM Comput. Surv. (CSUR) 2013, 45, 35. [CrossRef]
- 42. Vanhoucke, M.; Maenhout, B. NSPLib—A nurse scheduling problem library: A tool to evaluate (meta-)heuristic procedures. In Operational Research for Health Policy: Making Better Decisions, Proceedings of the 31st Annual Conference of the European Working Group on Operational Research Applied to Health Services; Peter Lang AG: Bern, Switzerland, 2007.