Evolutionary Techniques for the Nurse Scheduling Problem

Mehdi Sadeghilalimi, Malek Mouhoub[a] and Aymen Ben Said
Department of Computer Science, University of Regina, Regina, Canada

Keywords: Combinatorial Optimization, Nature-Inspired Techniques, Metaheuristics, Stochastic Optimization, Resource Allocation, Nurse Scheduling Problem (NSP).

Abstract: The Nurse Scheduling Problem (NSP) is a combinatorial optimization problem that creates weekly scheduling solutions for nurses. These solutions must satisfy constraints for the workload coverage requirements while optimizing one or more objectives related to hospital costs or nurses' preferences. Although exact methods may be used to solve the NSP and return the optimal solution, they usually come with an exponential time cost. Therefore, approximate methods may be considered as they offer a good trade-off between the quality of the solution and the running time. In this context, we propose a solving method based on Genetic Algorithms (GAs) to solve the NSP. To evaluate the efficiency of our proposed method, we conducted experiments on various NSP instances. Further, we compared the quality of the returned solutions against solutions obtained from exact methods and metaheuristics. The experimental results reveal that our proposed method can fairly compete with B&B in terms of the quality of the solution while delivering the solutions in much faster running times.

1 INTRODUCTION

The Nurse Scheduling Problem (NSP) is among the challenging NP-Hard combinatorial optimization problems in healthcare. The aim is to create weekly schedules that satisfy the workload constraints and simultaneously optimize some objectives. Various approaches were proposed to solve the NSP using exact and approximate methods. While exact methods guarantee the return of the optimal solution, they usually suffer from expensive running time costs. For this reason, researchers often look for alternatives to reduce this time complexity by exploring metaheuristics. The latter typically trade the quality of the solutions for better running times. In this context, we propose a new method based on the Genetic Algorithm (GA) to solve the NSP. To assess the performance of our proposed method, we adopted the NSP formulation from (Sadeghilalimi et al., 2023) and conducted multiple experiments using a set of NSP instances. Furthermore, we compared the obtained results against exact and metaheuristics. For the exact method, we used the Branch & Bound (B&B) algorithm from (Ben Said and Mouhoub, 2022) that relies on constraint propagation techniques (Dechter and Cohen, 2003) as a pre-processing step to remove locally inconsistent values. The latter step allows the reduction of the search space size before the execution of B&B. For the metaheuristics, we used variants of the Whale Optimization Algorithm (WOA) and Stochastic Local Search (SLS) (Sadeghilalimi et al., 2023). Note that the WOA variants correspond to different types of mutations we have considered for the exploration phase to ensure more diversification in the search and overcome local minima. We have adopted the same variants for our GA-based method. The SLS-based method starts by finding an initial solution using a depth-first search (DFS) technique and then attempts to tune it further to improve the solution's quality while maintaining feasibility. Similar to B&B, SLS uses constraint propagation to optimize the backtrack search. The experimental results are very promising as they reveal the efficiency of our proposed GA-based technique in providing a reasonable trade-off between the quality of the solution returned and the running time compared to the above-mentioned methods. Although B&B and DFS use constraint propagation before the search, the results show they still suffer from their expensive exponential time costs.

[a] https://orcid.org/0000-0001-7381-1064

Sadeghilalimi, M., Mouhoub, M. and Ben Said, A.
Evolutionary Techniques for the Nurse Scheduling Problem.
DOI: 10.5220/0012402300003639

In *Proceedings of the 13th International Conference on Operations Research and Enterprise Systems (ICORES 2024)*, pages 333-340
ISBN: 978-989-758-681-1; ISSN: 2184-4372

2 RELATED WORKS

In (Ben Said et al., 2021), the authors proposed an implicit approach to solve the NSP relying on Machine Learning (ML) algorithms to learn the frequent patterns and associations among past scheduling solutions. Although the proposed ML methods may generate solutions almost instantly, they come with a degree of uncertainty since the constraints and objectives are implicitly learned and represented through the learned patterns (e.g. association rules, trained ML models). In (Ben Said and Mouhoub, 2022), the authors proposed an explicit approach to solve the NSP using the Weighted Constraints Satisfaction Problem (WCSP) formalism (Larrosa, 2002; Bidar and Mouhoub, 2023; Lee and Leung, 2009) to overcome the uncertainty challenges. The authors formulated the NSP as a WCSP model and further solved it using B&B (we use this B&B algorithm as a comparative method for our proposed GA variants in the scope of this paper). B&B was also used in (Baskaran et al., 2014) to solve the NSP, and as discussed in (Woeginger, 2003), the challenge with exact algorithms like B&B is that they suffer from their exponential time cost, particularly when solving large-size problem instances.

Other related work relied on evolutionary methods based on metaheuristics to solve the NSP (Jan et al., 2000; Gutjahr and Rauner, 2007; Wu et al., 2013; Jafari and Salmasi, 2015; Rajeswari et al., 2017). These methods usually use random search to elicit candidate solutions while balancing between exploration and exploitation to escape local minima/maxima. The latter is an alternative to exact methods due to the time complexity challenge. However, returning the optimal solution is not guaranteed.

Hybrid methods that combine multiple algorithms were also explored to solve the NSP (Burke et al., 2001). For instance, in (Zhang et al., 2011), the authors proposed a hybrid method combining GAs and Variable Neighborhood Search (VNS). The GA was used to solve sub-problems and return initial feasible solutions that are consequently fed into the VNS to improve them. Experimental results show that the proposed method can return feasible solutions and therefore can be used effectively in solving other resource allocation problems.

3 PROBLEM FORMULATION

In the following we provide the Mixed-integer programming (MILP) formulation of the NSP. In addition, we also present the NSP modeling as a WCSP. The WCSP formulation is used by the B&B algorithm (Ben Said and Mouhoub, 2022) that we have considered in the comparative experiments reported in Section 6.

3.1 MILP Formulation

Table 1 lists the decision variables and all the required parameters needed for our formulation of the NSP. Basically, the main goal is to assign nurses to daily shifts[1] such that a set of constraints are met (following hospital personnel policies) while an overall cost is minimized.

Constraints

1. **Minimum and Maximum Number of Nurses per Shift.** The following constraint expresses the minimum and maximum assigned number of nurses per shift j during day k.

$$Q_{jk} \leq \sum_{i} x_{ijk} \leq S_{jk} \tag{1}$$

2. **Maximum Number of Shifts for a Given Nurse During the Schedule.** The following constraint sets the maximum number of shifts w_i, for a given nurse i during the schedule.

$$\sum_{j}\sum_{k} x_{ijk} \leq w_i \tag{2}$$

3. **Maximum Number of Consecutive Shifts.** The following constraint sets the maximum number of consecutive shifts L for a given nurse i during the schedule.

$$\sum_{k}^{d} \left(x_{i3k} + x_{i1(k+1 \, mod \, d)}\right) \leq L \tag{3}$$

4. **Maximum Number of Night Shifts.** Each nurse i should not work more than n_i night shifts in the schedule.

$$\sum_{k} x_{ijk} \leq n_i \qquad j = 3 \tag{4}$$

Objective: Hospital Costs to Minimize

$$\min\left(\sum_{i}\sum_{j}\sum_{k} c_{ij} \cdot x_{ijk}\right) \tag{5}$$

[1] We assume that all shifts have the same number of hours.

Table 1: Parameters of the NSP.

Parameters	Description
n	Number of nurses
i	Nurse index
j	A given shift: Morning shift ($j = 1$) Evening shift ($j = 2$) Night shift ($j = 3$)
d	The number of days in the schedule
c_{ij}	The cost of nurse i working in shift j for any day
x_{ijk}	The decision variable: 1, if nurse i works in shift j on day k, 0, otherwise
Q_{jk}	Minimum nurses needed for shift j in day k
S_{jk}	Maximum number of nurses required for shift j in day k
w_i	Maximum number of shifts during the schedule for nurse i
L	Maximum consecutive shifts that are allowed (a consecutive shift corresponds to $j = 3$ for a day k, followed by $j = 1$ for day $k+1 \mod d$)
n_i	Maximum night shifts during the schedule for nurse i

3.2 WCSP Formulation

The CSP is a powerful framework for modeling and solving combinatorial problems (Dechter and Cohen, 2003). Modeling a given problem using the CSP consists in formulating it in terms of a set of variables $X = \{x_1, \ldots, x_i, \ldots, x_n\}$, where each variable x_i is defined on a non-empty domain of possible values $dom(x_i$; and a set of constraints $C = \{c_1, ..., c_k\}$ restricting variables' assignment combinations. The goal of solving a CSP is to find a consistent assignment of values to all the variables from their domains such that all the constraints are satisfied. The WCSP (Larrosa, 2002; Bidar and Mouhoub, 2023; Lee and Leung, 2009) is an extension of the CSP which considers violation costs related to soft constraints or weights associated with variable domain values. In addition to finding a solution that satisfies all the constraints, the target of a WCSP is to optimize the solution's total cost.

More formally, a WCSP is defined by the tuple (X, D, C, K), where X, D, and C are variables, domains, and constraints respectively. K is the largest numerical cost value for for a given variable assignment.

Variables: $X = \{X_1, ..., X_n\}$, the set of nurses.
Domain: D is the set of shift patterns.
Constraints: $C = \{const_1, ..., const_4\}$

To elicit the NSP constraints, we rely on function $A(i, j, k, s)$ that returns 1 if Nurse i is assigned shift pattern j, and j covers shift s on day k, or 0 otherwise. Note that constraint $const_1$ is a global constraint that involves all the variables and constraints $const_2, const_3, const_4$ are unary constraints. For the WCSP formulation, we rely on the following parameters and indices to elicit the constraints.

Parameters and Indices:

n = Number of nurses
m = Number of possible shift patterns
c_{ij} = Cost of assigning nurse i the shift pattern j
q_{sk} = Minimum nurses needed for shift s in day k
p_{sk} = Maximum number of nurses required for shift s in day k
h_i = Maximum number of shifts for nurse i during the schedule
y = Maximum number of consecutive shifts (night shift followed by a morning shift)
b_i = Maximum number of night shifts for nurse i during the schedule
$i = \{1, .., n\}$: nurse index
$j = \{1, ..., m\}$: index of the weekly shift pattern
$k = \{1, .., 7\}$: day index
$s = \{1, .., 3\}$: shift index within a given day
$z = \{1, .., 21\}$: index of shifts in a shift pattern

Constraints:

1. **Minimum and Maximum Number of Nurses per Shift:** $const_1$

$$p_{sk} \leq \sum_{i=1}^{n} A(X_i, j_i, k, s) \geq q_{sk} \quad \forall k, \forall s, a_{j_i} \in D \quad (6)$$

2. **Maximum Number of Shifts for a Given Nurse During the Schedule:** $const_2$

$$\sum_{k=1}^{7}\sum_{s=1}^{3} A(X_i, j_i, k, s) \leq h_i \ \ \forall X_i, a_{j_i} \in D \quad (7)$$

3. **Maximum Number of Consecutive Shifts (Night Shifts Followed by Morning Shifts):** $const_3$

$$\sum_{k=1}^{6} A(X_i, j_i, k, 3) + A(X_i, j_i, k+1, 1) \leq y \ \ \forall X_i, a_{j_i} \in D \quad (8)$$

4. **Maximum Number of Night Shifts:** $const_4$

$$\sum_{k=1}^{7} A(X_i, j_i, k, 3) \leq b_i \ \ \forall X_i, a_{j_i} \in D \quad (9)$$

Soft Constraint:

$$f_i : a_{j_i} \in D \rightarrow c_{ij_i} \quad (10)$$

Objective: Hospital Costs to Minimize

$$Minimize(\sum_{i=1}^{n} c_{ij_i}) \ a_{j_i} \in D \quad (11)$$

4 PROPOSED SOLVING APPROACH

Our proposed method first enforces the satisfaction of all constraints. Then, the obtained consistent outputs are given to the GA algorithm to search for the optimal solutions.

More precisely, after generating a random population of potential solutions (schedules), the first part of the solving method consists in enforcing satisfiability by eliminating any detected constraint violation. After conducting a preliminary work, we came to the conclusion that constraints should be satisfied using the following order. First, constraint 4, which is related to the maximum number of night shifts, is satisfied. Then constraint 3, followed by constraint 2 are enforced. Finally, constraint 1, which is the most challenging constraint, is satisfied. To solve each of the above constraints, the algorithm first detects the variables involved in the constraint violations. Then, some of these variables are randomly selected, and their values are flipped in order to satisfy the related constraint. After enforcing the constraints as described earlier, the GA operators are applied on the feasible solutions to find the optimal one. In this context, the one-point crossover is first applied on some selected chromosomes. Then, mutation is conducted to ensure some diversity. Constraint satisfiability is enforced on the offsprings after each of these two operations. Figures 1 and 2 illustrate the crossover and mutation operations, respectively. In our example, we have 5 nurses working over 7 days. Like for the WOA (Sadeghilalimi et al., 2023), we consider the following variants for the mutation operator.

- **Random Resetting Mutation (RRM).** A number of entries of the NSP potential solution are randomly selected and their values are randomly changed. This process is depicted in Figure 2.
- **Swap Mutation (SwM).** After selecting a pair of variables, their values are swapped.
- **Scramble Mutation (ScM).** A subset of contiguous entries are selected from the potential solution. Then these values are randomly scrambled.
- **Inverse Mutation (IM).** A subset of contiguous entries are selected and inverted.

The fitness function corresponds to the NSP objective defined in Equation 5.

5 B&B, SLS, AND WOA FOR THE NSP

5.1 B&B

We adapted the B&B algorithm from (Ben Said and Mouhoub, 2022) to reflect the minimization variant of the NSP. B&B uses the Depth First Search (DFS) strategy to explore all candidate solutions and relies on the Upper Bound (UB) and Lower Bound (LB) parameters for pruning the non-optimal and infeasible solutions. In this context, the UB is used to record the cost of the best solution found during the search, and the LB is used to overestimate the quality of the solutions that may be obtained at any given node. The latter may be seen as a forward-checking technique; if the estimated LB is greater than the current UB, then there will be no need to continue exploring the current decision because it would definitely not lead to a better solution. Note that the LB is estimated by computing the real costs of already explored nodes in a given sub-branch plus the minimum costs of shift patterns that could possibly be assigned to the remaining variables/nurses. Although B&B guarantees the solution's optimality, it may come with an exponential time cost given the number of nurses and the domain size to be considered ($O(d^n)$, where d is the domain size and n the number of nurses). To overcome this limitation, constraint propagation (Dechter and Cohen, 2003) is used to reduce the search space and consequently minimize

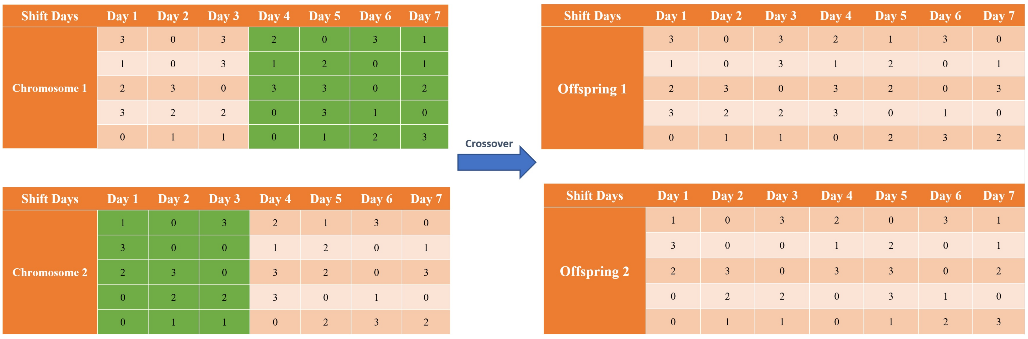

Figure 1: An example of the crossover operation.

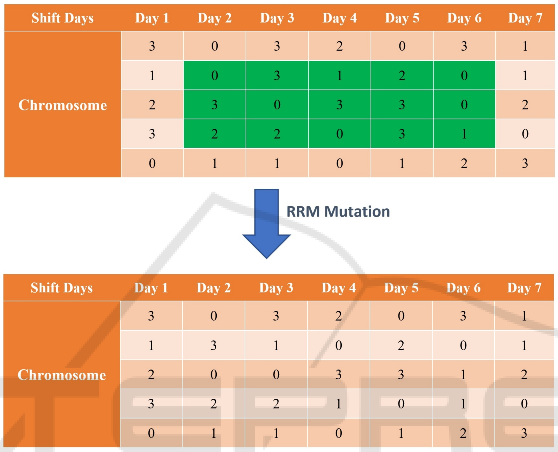

Figure 2: An example of the RRM Mutation.

the B&B execution time. Constraint propagation is also conducted as a pre-processing step before the execution of B&B to reduce the variable domain sizes by removing locally inconsistent values that violate unary, binary, and k-ary constraints. Enforcing constraint propagation may result in two different scenarios; the first one consists in proving the inconsistency of the problem in the case of removing all the values from a given variable domain (no solution exists). The second scenario corresponds to removing some of the values and obtaining a reduced-size domain which will requires less effort from the B&B algorithm to search for the optimal solution. We apply Node consistency (NC) (Larrosa, 2002) through the unary constraints 2, 3, and 4 (in Section 3), and we call B&B + NC, the search method using NC as a preprocessing phase before running B&B. Furthermore, we apply the Generalized Arc Consistency (GAC) algorithm (Lecoutre and Szymanek, 2006; Cheng and Yap, 2010) through global constraint 1 (in Section 3) to eliminate inconsistent domain values that are not part of any feasible solution, and we call, B&B + NC + GAC, the method using both NC and GAC as a preprocessing step before running B& B.

5.2 SLS

We consider SLS variants: SLS, DFS+SLS, and DFS + NC + GAC + SLS. These variants work by obtaining an initial solution and then trying to enhance it by sequentially looking for a better value substitution for each variable while maintaining the solution's feasibility. The difference between the three variants is the method used to get the initial solution. For SLS, the initial solution is obtained using a random search. In DFS+SLS, the initial solution is found after running a Depth-First-Search (DFS) and finding the first feasible solution. Finally, for DFS + NC + GAC + SLS, the initial solution is found following a DFS search after enforcing constraint propagation (NC and GAC) as a preprocessing step to eliminate inconsistent domain values and tackle the exponential time cost that may come from DFS.

5.3 WOA

The WOA that we have adopted (Sadeghilalimi et al., 2023) is an adaptation of the original (Mirjalili and Lewis, 2016) for the NSP. WOA is inspired by the behavior of humpback whales and can be seen as a combination of both the moth flame and the grey wolf techniques (Camacho-Villalón et al., 2022). In WOA, each whale corresponds to a chromosome as illustrated in Figures 1 and 2. Exploration is performed with random whales movements through RRM, ScM, SwM, and IM mutations. Exploitation is Achieved by having each whale X move toward the best whale X^* through **shrinking encircling** and **spiral motions**. In the case of the NSP, these operators are defined as follows.

Shrinking Encircling

$$D = |C \cdot X^*(t) - X(t)| \tag{12}$$

$$X(t+1) = X^*(t) - A \cdot D \tag{13}$$

$$A = 2a \cdot r - a \tag{14}$$

$$C = 2 \cdot r \tag{15}$$

a and r are random parameters in [0,2] and [0,1] respectively. C is set to 1. Equation 13 will then allow whale X to move closer to X^* by reducing (according to A) the number of entries that are different in both whales (using the Hamming distance).

Spiral Attack

Each whale (represented by $X(t)$) approaches its prey (best whale, X^*) by following a spiral curve. b is a constant and l is a random variable between [-1, 1].

$$X(t+1) = D \cdot e^{bl} \cdot \cos(2\pi l) + X^*(t) \tag{16}$$

$$X(t+1) = X^*(t) - A \cdot D^{'} \tag{17}$$

$$A = e^{bl} \cdot \cos(2\pi l) \tag{18}$$

To balance shrinking and spiral attacks, a random parameter, p, is generated between [0, 1] to choose between the two attacks as follows.

$$A = \begin{cases} 2a \cdot r - a & p < 0.5 \\ e^{bl} \cdot \cos(2\pi l) & p \geq 0.5 \end{cases} \tag{19}$$

6 EXPERIMENTATION

To evaluate the performance of our proposed GA-based method, we conducted comparative experiments against variants of B&B (Ben Said and Mouhoub, 2022) and metaheuristics (SLS and WOA) (Sadeghilalimi et al., 2023). All the algorithms are implemented in MATLAB software on a computer with a Intel Core i5-6200U processor at 2.3 GHz and 8 GB of RAM. The NSP instances parameters and cost functions are depicted in Tables 2 and 3. The parameters guiding the techniques we used in the experiments are tuned, to their best using Chess Rating System (CRS-Tuning) (Veček et al., 2016). While CRS-Tuning is comparable to other known tuning methods such as Revac (Nannen and Eiben, 2007) and F-Race (Birattari et al., 2002), it has several advantages as reported in (Veček et al., 2016). For WOA, a and r are randomly selected from $[0,1]$ while l is randomly selected from $[-1,+1]$. In the case of our proposed GA method, a one-point crossover is chosen with probability 0.8, and the roulette wheel method is adopted. For the mutation, we consider Swap Mutation, Sequence Swap Mutation, and Inversion Mutation, respectively with probability 0.1, 0.02, and 0.08.

Table 2: The NSP parameters used in the experiments.

n	d	Q_{jk}	S_{jk}	w_i	L	n_i
5	7	1	4	5	2	3
10	7	1	7	5	2	3
15	7	1	12	5	2	3
20	7	1	15	5	2	3
30	7	1	25	5	2	3
50	7	1	35	5	2	3
60	7	1	45	5	2	3
80	7	1	65	5	2	3

Table 3: The cost table.

Nurse no.	$c(i,j)$		
	Shift 1	**Shift 2**	**Shift 3**
1	0.81	0.16	0.64
2	0.90	0.79	0.37
3	0.12	0.31	0.81
4	0.91	0.52	0.53
5	0.63	0.16	0.35
6	0.09	0.60	0.93
7	0.27	0.26	0.87
8	0.54	0.65	0.55
9	0.95	0.68	0.62
10	0.96	0.74	0.58
11	0.15	0.45	0.20
12	0.97	0.08	0.30
13	0.95	0.22	0.47
14	0.48	0.91	0.23
15	0.80	0.22	0.84

We have adopted the same population size for both GAs and WOA. This is justified by the fact that the function evaluation (fitness computation) cost is the most expensive part in a nature-inspired technique, and its related running time is directly affected by the number of individuals in a population. Table 4 reports on the experiment results comparing variants of the

Table 4: Comparative running time and quality of the solution for all the methods.

Method	Number of Nurses															
	5		10		15		20		30		50		60		80	
	BS	RT (s)	BS	RT (s)	BS	RT (s)	BS	RT (s)	BS	RT (s)	BS	RT (s)	BS	RT (s)	BS	RT (s)
GA + RRM	8.87	3.85	16.95	6.77	22.92	8.32	34.74	11.15	49.95	21.95	88.84	150.66	**114.38**	285.57	**156.06**	225.70
GA + SwM	9.01	6.31	17.06	6.82	24.58	8.45	35.19	11.57	51.2	22.01	89.12	189.72	114.99	298.75	156.89	331.64
GA + ScM	9.45	6.68	17.20	7.11	25.23	9.22	35.20	11.89	54.70	22.48	89.65	214.43	115.58	312.71	157.69	350.14
GA + IM	9.51	7.03	17.25	8.32	25.64	10.18	35.43	12.89	54.97	22.84	90.15	205.98	116.11	322.44	158.67	358.15
WOA	10.22	1.01	21.29	1.62	33.39	3.03	40.83	2.90	69.93	37.86	106.25	244.34	124.75	2221.78	177.66	3393.23
WOA + RRM	10.82	1.50	21.72	1.46	30.56	1.50	43.09	9.98	65.04	44.33	105.04	412.05	129.19	653.03	170.18	4196.58
WOA + SwM	10.29	0.95	22.81	1.53	32.48	**0.97**	41.85	14.96	65.28	24.40	103.26	340.61	127.09	1073.41	175.55	959.06
WOA + ScM	9.78	1.73	19.56	**0.11**	29.51	3.49	40.68	6.01	63.38	42.05	102.01	332.04	123.40	240.24	169.48	1301.61
WOA + IM	10.45	1.85	22.19	0.21	33.01	1.60	41.21	15.41	63.49	4.14	103.78	277.68	127.66	599.18	173.49	2005.55
SLS	11.57	**0.69**	24.10	0.50	32.99	1.23	47.28	**1.12**	72.32	**1.17**	109.49	**1.49**	142.92	**1.97**	189.89	**2.79**
DFS + SLS	14.86	18.90	28.68	164.11	38.91	345.57	43.62	404.75	76.08	1060.31	119.21	3873.79	148.34	4351.57	189.96	6901.90
DFS + NC + GAC + SLS	12.34	5.49	25.81	89.24	32.28	100.16	49.33	246.71	67.98	1394.28	109.96	3830.05	140.86	4781.31	185.54	5813.59
B&B + NC	**9.68**	1894	**18.86**	14415	**25.55**	22689	**38.58**	28137	**52.34**	34259	**89.35**	41459	-	-	-	-
B&B + NC + GAC	**9.68**	534	**18.86**	11400	**25.55**	16211	**38.58**	23418	**52.34**	29768	**89.35**	37108	-	-	-	-

Table 5: Comparative quality of the solution in Best, Average, and Standard deviation.

Method	Number of Nurses																							
	5			10			15			20			30			50			60			80		
	Best	Ave.	Dev.	Best	Ave.	Dev.	Best	Ave.	Dev.	Best	Ave.	Dev.	Best	Ave.	Dev.	Best	Ave.	Dev.	Best	Ave.	Dev.	Best	Ave.	Dev.
GA + RRM	**8.87**	10.12	0.95	**16.95**	18.12	1.12	**22.92**	24.01	1.42	**34.74**	36.21	1.39	**49.95**	51.54	1.51	**88.84**	90.27	1.61	**114.38**	116.30	1.67	**156.06**	158.39	1.49
GA + SwM	9.01	10.28	1.09	17.06	18.53	1.53	24.58	25.93	1.42	35.19	36.33	1.54	51.2	52.56	0.61	89.12	91.29	1.13	114.99	116.81	1.08	156.89	158.84	1.01
GA + ScM	9.45	10.37	0.73	17.20	18.79	0.64	25.23	26.39	0.59	35.20	36.77	0.46	54.70	56.22	1.23	89.65	93.02	1.75	115.58	117.11	1.32	157.69	159.55	1.18
GA + IM	9.51	10.85	0.69	17.25	18.74	0.64	25.64	27.2	1.08	35.43	37.14	1.82	54.97	56.89	1.49	90.15	93.41	1.81	116.11	118.21	1.82	158.67	161.01	1.48
WOA	10.22	11.79	0.76	21.29	22.33	0.88	33.39	35.21	1.45	40.83	42.25	1.77	69.93	71.09	1.31	106.25	109.2	1.75	124.75	126.5	1.4	177.66	179.29	0.91
WOA + RRM	10.82	11.28	0.69	21.72	23.01	1.93	30.56	32.15	1.2	43.09	44.98	1.38	65.04	67.25	1.39	105.04	107.34	2.18	129.19	131.35	1.67	170.18	172.71	1.48
WOA + SwM	10.29	11.89	1.54	22.81	24.51	1.45	32.48	34.33	1.69	41.85	43.39	1.67	65.28	68.01	2.18	103.26	105.21	1.88	127.09	129.92	2.22	175.55	178.18	1.49
WOA + ScM	9.78	11.15	1.4	19.56	21.57	1.27	29.51	31.35	2.13	40.68	42.59	1.69	63.38	65.83	1.86	102.01	105.2	1.12	123.40	125.35	2.58	169.48	171.96	1.46
WOA + IM	10.45	11.87	1.58	22.19	23.94	1.88	33.01	35.13	1.42	41.21	43.84	1.38	63.49	65.42	1.04	103.78	105.86	2.27	127.66	129.63	1.84	173.49	175.83	1.14
SLS	11.57	14.44	1.33	24.10	26.81	1.48	32.99	38.09	2.28	47.28	51.45	2.03	72.32	77.75	2.49	109.49	122.74	4.67	142.92	147.69	3.67	189.89	202.84	6.91

GA and WOA algorithms with B&B and SLS, as described previously. The quality of the best solution returned (BS) and the corresponding running time (RT) were used as comparison criteria. All the results are averaged over 20 runs. The experiments were conducted on several NSP instances, with the number of nurses varying from 5 to 80. Given that it is an exact method, B& B returns the optimal solution for each problem instance, up to 50 nurses. When the number of nurses exceeds 50, B & B fails to return the optimal solution due to its exponential time cost. For these large instances, GA with RRM is the best method regarding the returned solution's quality. The same remark regarding the exponential time cost can be said when adding DFS as an initial step for the SLS algorithm. In both B&B and SLS, constraint propagation does help lower the running time (as a consequence of reducing the running time). However, this effort is insufficient to compete with WOA and GA variants. SLS is the best method in terms of running time. However, the quality of the solution returned by the latter is inferior to those returned by variants of GA and WOA. Given that our experiments are conducted over 20 runs, we conducted a statistical analysis on the solution returned by each metaheuristic. We reported the results in Table 5 regarding the quality of the returned solution.

7 CONCLUSION AND FUTURE WORK

We have proposed a GA-based method to solve the NSP. To evaluate the efficiency of our method, we conducted experiments against exact and approximate techniques. The obtained results are promising compared to B&B, which suffers from its exponential time cost and fails in returning solutions for large-size problem instances.

In the near future, we plan to explore other nature-inspired techniques (Hmer and Mouhoub, 2016; Korani and Mouhoub, 2021) including Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 1995) and further plan to tackle the dynamic variant of the NSP problem, which is characterized by the unpredictable occurrence of events such as nurses calling in sick or sudden changes in hospital demand in particular scenarios (such as emergencies) (Bidar and Mouhoub, 2022; Mouhoub, 2003). We will also use variable ordering heuristics (Yong and Mouhoub, 2018) to improve the time efficiency of the B& B algorithm.

REFERENCES

Baskaran, G., Bargiela, A., and Qu, R. (2014). Integer programming: Using branch and bound to solve the nurse scheduling problem. In *2014 International Conference on Artificial Intelligence and Manufacturing Engineering (IIE ICAIME2014)*.

Ben Said, A., Mohammed, E. A., and Mouhoub, M. (2021). An implicit learning approach for solving the nurse scheduling problem. In Mantoro, T., Lee, M., Ayu, M. A., Wong, K. W., and Hidayanto, A. N., editors, *Neural Information Processing*, pages 145–157, Cham. Springer International Publishing.

Ben Said, A. and Mouhoub, M. (2022). A constraint satisfaction problem (csp) approach for the nurse scheduling problem. In *2022 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 790–795.

Bidar, M. and Mouhoub, M. (2022). Nature-inspired techniques for dynamic constraint satisfaction problems. *Operations Research Forum*, 3(2):1–33.

Bidar, M. and Mouhoub, M. (2023). Nature-inspired algorithms for solving weighted constraint satisfaction problems. In Rocha, A. P., Steels, L., and van den Herik, H. J., editors, *Proceedings of the 15th International Conference on Agents and Artificial Intelligence, ICAART 2023, Volume 3, Lisbon, Portugal, February 22-24, 2023*, pages 63–72. SCITEPRESS.

Birattari, M., Stützle, T., Paquete, L., Varrentrapp, K., et al. (2002). A racing algorithm for configuring metaheuristics. In *Gecco*, volume 2.

Burke, E., Cowling, P., De Causmaecker, P., and Berghe, G. V. (2001). A memetic approach to the nurse rostering problem. *Applied intelligence*, 15(3):199–214.

Camacho-Villalón, C. L., Dorigo, M., and Stützle, T. (2022). Exposing the grey wolf, moth-flame, whale, firefly, bat, and antlion algorithms: six misleading optimization techniques inspired by bestial metaphors. *International Transactions in Operational Research*.

Cheng, K. C. and Yap, R. H. (2010). An mdd-based generalized arc consistency algorithm for positive and negative table constraints and some global constraints. *Constraints*, 15(2):265–304.

Dechter, R. and Cohen, D. (2003). *Constraint processing*. Morgan Kaufmann.

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

Hmer, A. and Mouhoub, M. (2016). A multi-phase hybrid metaheuristics approach for the exam timetabling. *International Journal of Computational Intelligence and Applications*, 15(04):1650023.

Jafari, H. and Salmasi, N. (2015). Maximizing the nurses' preferences in nurse scheduling problem: mathematical modeling and a meta-heuristic algorithm. *Journal of industrial engineering international*, 11(3):439–458.

Jan, A., Yamamoto, M., and Ohuchi, A. (2000). Evolutionary algorithms for nurse scheduling problem. In *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No. 00TH8512)*, volume 1, pages 196–203. IEEE.

Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, volume 4, pages 1942–1948. IEEE.

Korani, W. and Mouhoub, M. (2021). Review on nature-inspired algorithms. In *Operations research forum*, volume 2, pages 1–26. Springer.

Larrosa, J. (2002). Node and arc consistency in weighted csp. In *AAAI/IAAI*, pages 48–53.

Lecoutre, C. and Szymanek, R. (2006). Generalized arc consistency for positive table constraints. In *International conference on principles and practice of constraint programming*, pages 284–298. Springer.

Lee, J. H. and Leung, K. L. (2009). Towards efficient consistency enforcement for global constraints in weighted constraint satisfaction. In *Twenty-First International Joint Conference on Artificial Intelligence*.

Mirjalili, S. and Lewis, A. (2016). The whale optimization algorithm. *Advances in engineering software*, 95:51–67.

Mouhoub, M. (2003). Dynamic path consistency for interval-based temporal reasoning. In *Applied Informatics 2003*, pages 393–398.

Nannen, V. and Eiben, A. E. (2007). Efficient relevance estimation and value calibration of evolutionary algorithm parameters. In *2007 IEEE congress on evolutionary computation*, pages 103–110. IEEE.

Rajeswari, M., Amudhavel, J., Pothula, S., and Dhavachelvan, P. (2017). Directed bee colony optimization algorithm to solve the nurse rostering problem. *Computational intelligence and neuroscience*, 2017.

Sadeghilalimi, M., Mouhoub, M., and Said, A. B. (2023). Solving the nurse scheduling problem using the whale optimization algorithm. In Dorronsoro, B., Chicano, F., Danoy, G., and Talbi, E.-G., editors, *Optimization and Learning*, pages 62–73, Cham. Springer Nature Switzerland.

Veček, N., Mernik, M., Filipič, B., and Črepinšek, M. (2016). Parameter tuning with chess rating system (crs-tuning) for meta-heuristic algorithms. *Information Sciences*, 372:446–469.

Woeginger, G. J. (2003). Exact algorithms for np-hard problems: A survey. In *Combinatorial optimization—eureka, you shrink!*, pages 185–207. Springer.

Wu, J.-j., Lin, Y., Zhan, Z.-h., Chen, W.-n., Lin, Y.-b., and Chen, J.-y. (2013). An ant colony optimization approach for nurse rostering problem. In *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pages 1672–1676. IEEE.

Yong, K. W. and Mouhoub, M. (2018). Using conflict and support counts for variable and value ordering in csps. *Applied Intelligence*, 48:2487–2500.

Zhang, Z., Hao, Z., and Huang, H. (2011). Hybrid swarm-based optimization algorithm of ga & vns for nurse scheduling problem. In *International Conference on Information Computing and Applications*, pages 375–382. Springer.