Applied Mathematics and Computational Intelligence
Volume 15, No. 1, 2026 [119 – 136]

Development of A Nurse Scheduling Timetable in The Emergency Department Using Bat Algorithm

Muhammad Haiman Mohd Haizani[1], Nor Aliza AB Rahmin[2*], Risman Mat Hasim[3]

[1,2*,3]Department of Mathematics and Statistics, Faculty of Science, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia.

* Corresponding author: aliza@upm.edu.my

Received: 6 May 2025
Revised: 28 December 2025
Accepted: 28 January 2026

ABSTRACT

Nurse scheduling in the emergency department (ED) is a complex optimisation problem that requires balancing nurse workload while adhering to operational constraints. This study applies the Bat Algorithm (BA), a metaheuristic optimisation technique, to generate efficient nurse schedules. The proposed approach aims to minimise deviations from workload constraints, reduce soft constraint violations, and ensure adequate rest periods while avoiding undesirable shift transitions. The BA was implemented in MATLAB and evaluated against traditional Goal Programming (GP) methods. Experimental results demonstrated its efficiency, with a computational time of 7.67 seconds. The BA outperformed GP by achieving lower deviations, indicating fewer unnecessary shifts and transitions that balance the workload for nurses while improving nurse satisfaction and operational efficiency. These findings highlight the potential of BA to enhance workforce management in healthcare settings and contribute to improved patient care.

Keywords: Bat Algorithm, Emergency Department, Metaheuristic, Scheduling.

1 INTRODUCTION

The emergency department (ED) is one of the busiest and most demanding hospital units, characterised by unpredictable patient inflows and a need for rapid, high-quality medical care. Effective patient flow management ensures timely clinical decision-making and appropriate treatment. Upon arrival, patients undergo initial assessments, including streaming, triage, and rapid assessment, to prioritise critical cases and initiate prompt treatment.

Managing ED staff scheduling, particularly for nurses, is crucial to maintaining operational efficiency and delivering high-quality patient care. Nurses play a vital role in supporting physicians and addressing patient needs. However according to [1], high workloads and irregular shifts can lead to burnout, job dissatisfaction, and increased turnover, then affecting patient outcomes. A well-designed nurse schedule must balance workload distribution, ensure adequate rest periods, and comply with operational constraints.

Nurse scheduling involves satisfying hard constraints, such as shift coverage and legal working hours, while optimising soft constraints, like nurse preferences and workload balance. Traditional manual scheduling methods often struggle to meet these requirements, leading to the adoption of advanced computational techniques. Metaheuristic algorithms, including Genetic Algorithm (GA), Bat Algorithm (BA), Tabu Search, Artificial Bee Colony (ABC), and Particle Swarm Optimisation (PSO), offer practical solutions by efficiently navigating complex search spaces to generate high-quality schedules.

This study explores the application of metaheuristic algorithms to the nurse scheduling problem (NSP) in the ED. By optimising schedules, while addressing both hard and soft constraints, these methods aim to reduce nurse workload imbalances, enhance job satisfaction, and improve ED operational efficiency, then contributing to better patient care [2].

2 LITERATURE REVIEW

Effective scheduling is a critical challenge in healthcare, covering staff, patients, and resource allocation. Various studies have explored optimising techniques to improve scheduling efficiency. For instance, home healthcare scheduling has been addressed using Adaptive Large Neighbourhood Search (ALNS) and matheuristics to prioritise patient visits, with ALNS proving to be faster for smaller tasks and matheuristics offering better accuracy for larger ones [3]. Similarly, hospital operating room (OR) scheduling has been optimised using genetic algorithms and fuzzy models to balance capacity utilisation, reducing delays in patient admissions and discharge [4].

In the ED, overcrowding remains a significant issue, often leading to increased nurse workload and inefficient patient flow. Strategies such as discrete event simulation and capacity staff roles have been proposed to optimise bed management and reduce patient waiting times [5]. Nurse scheduling, in particular, is crucial because it ensures the delivery of high-quality patient care while preventing staff burnout. Several optimisation approaches have been introduced, including Goal Programming (GP)[6], mixed-integer programming, and heuristic methods, to improve scheduling fairness, efficiency, and job satisfaction [7, 8]. Studies have demonstrated that well-optimised nurse schedules reduce shift imbalances, minimise consecutive work hours, and enhance overall staff performance [9, 10].

Heuristic and metaheuristic algorithms have been widely applied to healthcare scheduling, including nurse rostering. Techniques such as tabu search, genetic algorithms, and PSO have shown promising results in balancing workload distribution and meeting hospital constraints [11–13]. The BA, in particular, has emerged as a strong contender for solving NSPs due to its ability to efficiently handle large and complex constraints [14]. This study focuses on applying BA to optimise nursing scheduling in the ED, aiming to enhance operational efficiency, improve nurse satisfaction, and ensure high-quality patient care.

3 METHODOLOGY

We refer to several past studies on the BA, including its performance, inspiration, and the algorithm itself, as discussed in [15]. Additionally, we refer to the study by [14], which applied BA for scheduling nurses in a radiology department. The results of that study indicate that BA outperforms other methods, such as PSO.

3.1 Original Bat Algorithm

From [15], the BA is inspired by the echolocation behaviour of microbats, which use sound pulses to detect prey and navigate them. In BA, candidate solutions are represented as bats that adjust their positions based on pulse emission, frequency modulation, and loudness. These mechanisms balance exploration and exploitation in the search space.

Mathematically, BA updates each bat's position and velocity using frequency-dependent rules influenced by the best-known solution. The algorithm also incorporates a local search mechanism, where solutions are refined through small adjustments based on loudness and random perturbations. Over time, loudness decreases while the pulse emission rate increases, allowing the search to focus on promising regions.

The iterative process of BA involves updating solutions while adapting algorithmic parameters to improve optimisation. Studies, such as [15], have shown that BA outperforms methods like PSO and Harmony Search (HS) in both accuracy and efficiency. Further refinements, such as time delay metrics, can enhance BA's adaptability in complex optimisation problems.

3.2 Bat Algorithm for Schedule

The BA, as applied in [14], was used to schedule radiological technologists in a hospital setting. The study demonstrated that BA, when combined with heuristic methods, effectively generates feasible nurse schedules. Each bat in the algorithm represents a candidate solution $x = (x_1, x_2, ..., x_d)$, where d is the number of bats, while the objective function $F(x)$ evaluates the schedule's quality. Figure 1 illustrates the flowchart for the BA.

3.2.1 Initialisation and Constraint Handling

The algorithm initialises bats with random values for position (x_i^0), velocity (v_i^0), loudness (A_i^0), pulse emission rate (r_i^0), and frequency (f_i^0), where frequency is sampled from $[f_{\min}, f_{\max}]$. A decision tree method ensures hard constraints are satisfied, with a correction mechanism applied to repair infeasible schedules.

3.2.2 Searching and Solution Updates

The search process begins with $t = 1$ and iteratively updates each bat's frequency, velocity, and position based on:

$$f_i^t = f_{\min} + (f_{\max} - f_{\min})\beta, \quad \beta = U(0,1), \tag{1}$$
$$v_i^t = v_i^{t-1} + x_* - x_i^t \cdot f_i^t, \tag{2}$$
$$x_i^t = x_i^{t-1} + v_i^t. \tag{3}$$

If the condition rand $> r^t$ is met, local exploration is performed using:

$$x_{\text{new}} = x_{\text{old}} + \varepsilon A_i^t, \quad \varepsilon = U(-1,1). \tag{4}$$

Local exploration includes generating neighbouring solutions and using a greedy search algorithm to improve results. Bats may also perform random searches to enhance diversity.

3.2.3 Acceptance Criteria and Convergence

Generated schedules are validated against hard constraints, with necessary repairs applied. If a new solution improves the best-known solution ($F(x_i) < F(x_*)$) and satisfies rand $< A^t$, the bat updates its position. The loudness and pulse emission rate evolve as follows:

$$A_i^{t+1} = \alpha A_i^t, \quad \alpha \in \text{Uniform}[0,1], \tag{5}$$
$$r_i^t = r_i^0[1 - \exp(-\gamma t)], \quad \gamma = \text{positive constant}. \tag{6}$$

This model simulates the behaviour of bats converging towards an optimal solution.

3.2.4 Termination and Optimal Schedule Selection

Solutions are ranked based on objective function values, with x_* updated accordingly. The algorithm terminates when a predefined condition is met, such as reaching a set iteration limit or achieving a solution comparable to that of IBM ILOG CPLEX. The best bat x_* is returned as the optimal nurse schedule.

3.3 Modified Bat Algorithm Method

The BA for nurse scheduling consists of two primary stages: generating initial bats and performing the search procedure. The algorithm is tailored to accommodate the constraints and requirements of scheduling nurses in an emergency department. To generate the initial population (initial bat), instead of randomly assigning nurses to the schedule, which may result in violations of hard constraints, we adopt a more controlled approach. For each shift on a given day, a nurse is randomly selected and assigned to that shift. Once assigned, the nurse is removed from the pool of available nurses for that day to ensure that no nurse is assigned to more than one shift per day. This approach inherently satisfies the constraint that a nurse can work only one shift per day. Additionally, the number of nurses required for each shift on each day is determined randomly, adding diversity to

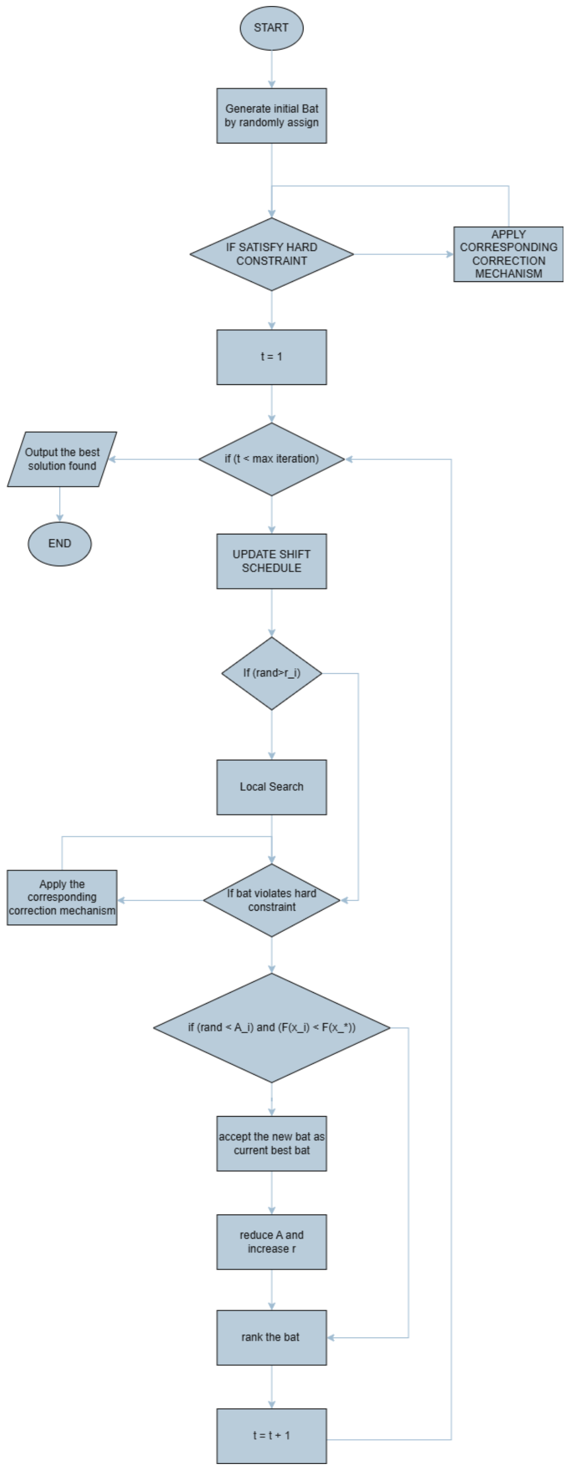

Figure 1 : Flowchart for the Bat Algorithm's Schedule

the generated schedules while maintaining feasibility. This process will be discussed in greater detail in the following section.

3.3.1 Generating Initial Bats

At the initialisation stage, a population of bats is created, where each bat represents a potential solution $x = (x_1, x_2, ..., x_d)$ to the NSP. A three-dimensional array of size (numNurse $\times$ numShift $\times$ numDays) is initialised with zeros. A list of 15 available nurses is then generated. From this list, five nurses are randomly selected and assigned to Shift 1 (morning shift). Once assigned, these nurses are removed from the available pool. The required number of nurses for Shift 2 and Shift 3 is then selected from the remaining nurses, ensuring that no nurse is scheduled for more than one shift per day. This process is repeated for all days until a complete schedule is formed. Table 1 presents the pseudocode for this procedure, and Figure 2 illustrates the corresponding flowchart.

Table 1 : Pseudo Code for Generating the Initial Schedule

```
Generate Initial Bats
Initialize a 3D matrix filled with zeros.
Initialize a list of available nurses.
for each day = 1 until total days
    for morning, evening, and night shifts
        Randomly select five nurses from available nurses and assign them to the morning shift.
        Remove assigned nurses from the available list.
        Randomly select five nurses from available nurses and assign them to the evening shift.
        Remove assigned nurses from the available list.
        Randomly select three nurses from available nurses and assign them to the night shift.
end
```

3.3.2 Bat Algorithm Parameter Initialisation

The objective function $F(x)$ evaluates the quality of each generated schedule by minimising constraint violations. Each bat is assigned a position (x_i^0), velocity (v_i^0), loudness (A_i^0), pulse emission rate (r_i^0), and frequency (f_i^0). The initial position x_i^0 is derived from the previously generated schedule. The velocity (v_i^0) is initially set to zero, loudness (A_i^0) is set to 1, and pulse emission rate (r_i^0) is set to zero. The frequency f_i^0 is randomly selected from a predefined range $[f_{\min}, f_{\max}]$ to balance exploration and exploitation during the search process.

3.3.3 Feasibility Check and Solution Correction

Each generated schedule undergoes a feasibility check to ensure that it satisfies the hard constraints, such as maintaining the minimum number of nurses required per shift. If a bat violates any constraints, a correction mechanism is applied to modify the schedule while preserving feasibility. The corrected schedules are then used in subsequent iterations to refine the solution.

3.3.4 Search Procedure

The search phase begins with $t = 1$ and continues until the maximum iteration count ($t = 400$) is reached. During each iteration, the algorithm updates the frequency, velocity, and position of each bat according to the equations 1, 2, and 3. If the pulse emission condition rand $> r_t$ is met, a local search is performed by generating a nearby solution around the bat's current position using equation 4. If generated solutions are validated against hard constraints, then necessary repair mechanisms are applied. If a new solution satisfies the acceptance criteria rand $< A_t$ and improves upon the current best ($F(x_i) < F(x_*)$), the bat updates its position. Loudness A_i^t decreases according to equation 5, and the pulse emission rate r_i^t increases according to equation 6, simulating the bats' convergence toward an optimal solution.

3.3.5 Termination and Selection of Optimal Schedule

The algorithm ranks all solutions based on their objective function values, updating x_* accordingly. The process continues until the stopping criterion is met, which, in this case, is the maximum iteration count. The algorithm then outputs x_*, representing the best nurse schedule. The complete pseudocode for the nurse scheduling algorithm is presented in Table 2 and Figure 2.

Table 2 : Pseudocode for the Modified BA Method

Generating Initial Bats
Generate initial bats. Each bat represents by $\mathbf{x} = (x_1, x_2, ..., x_d)^T$. The objective function value of a bat is $F(x)$, where $\mathbf{x}_*$ is the current bat.
Parameter initialisation for each bat: position $\mathbf{x_i^0}$, velocity $\mathbf{v_i^0}$, and loudness $\mathbf{A_i^0}$, pulse rate $\mathbf{r_i^0}$, and frequency f_i^0; Here, the pulse frequency f falls in a range of $[f_{min}, f_{max}]$; f_{min} is the minimal pulse frequency; and f_{max} is the maximal pulse frequency.
for each bat in the population
if the current bat violate any one of the hard constraints
Apply the corresponding correction mechanism.
end if
end for
Output the feasible solution of each bat;
Searching Procedures
$t = 1$;
Loop
for each bat in the population at the t iteration
Update the shift schedule for each bat, $\mathbf{x_i^t}$, by using equation (1), (2), and (3)
if $(rand > r_i)$ at the t iteration
Generate an adjacent bat for a local search by using equation (4)
end if
if the current bat violates any one of the hard constraints
Apply the corresponding shift schedule correction mechanisms to repair the current bat;
end if
if $(rand < A_i)$ and $(F(\mathbf{x_i}) < \mathbf{F}(\mathbf{x}_*))$ at the t iteration
Accept the new bat $(\mathbf{x_i^t})$ as the current best bat $(\mathbf{x}_*)$;
Reduce A_i^t and increase r_i^t by using Equation (5) and (6)
end if
Rank the bats' solutions by objective function values and update the current best bat $(\mathbf{x}_*)$;
end for
$t = t + 1$;
end Loop
Output the current best bat $(\mathbf{x}_*)$ as the best solution.

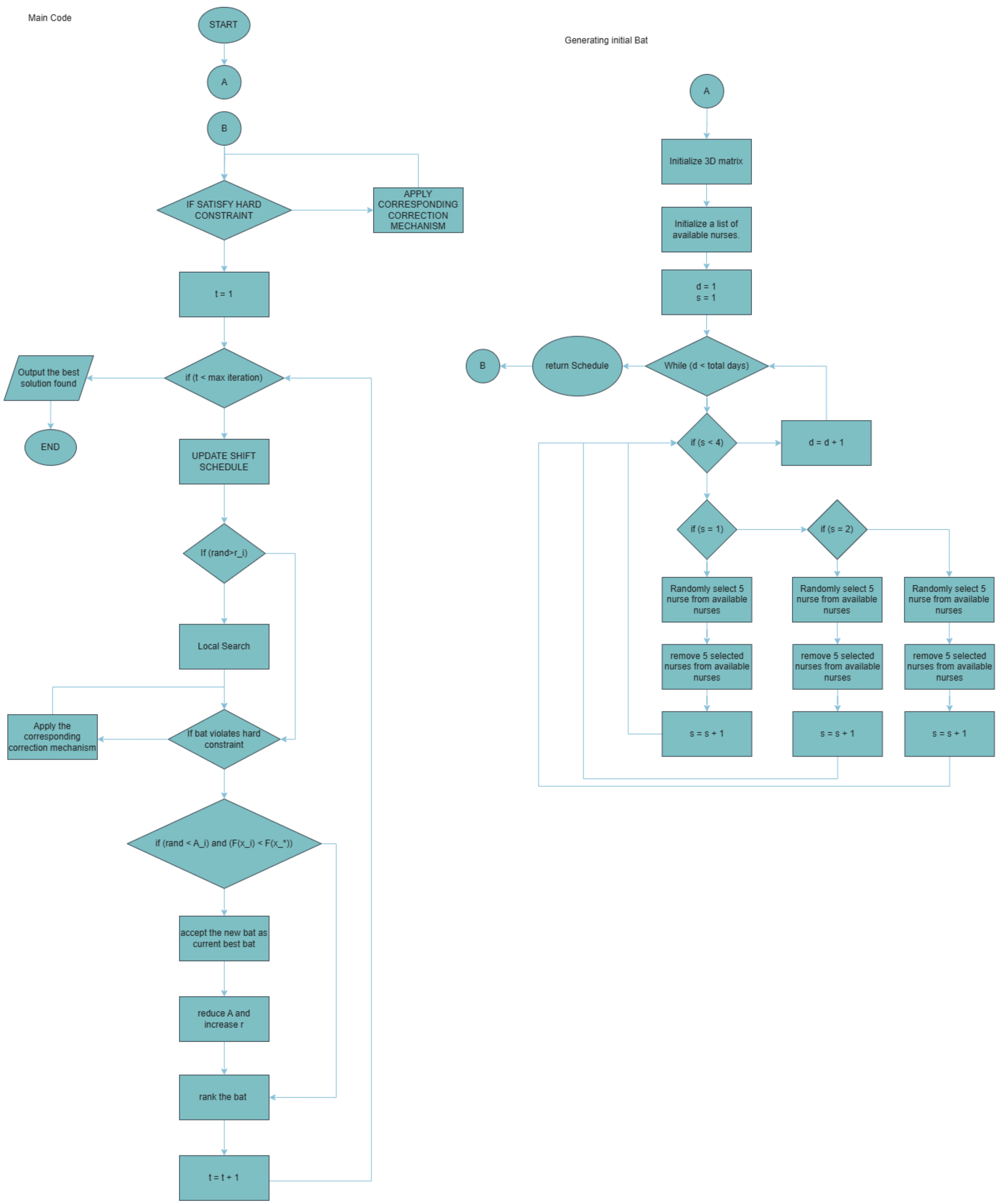

Figure 2 : Flowchart for the Modified Bat Algorithm (BA)

4 NURSE SCHEDULING PROBLEM (NSP) IN EMERGENCY DEPARTMENT (ED)

4.1 Mathematical Model

The optimisation model described in this section determines the hard and soft constraints considered when scheduling nurses in the ED. In this study, the model from [7] was considered to solve the problem.

4.1.1 Notation

Table 3 defines the notations used in the nurse scheduling model. The parameters include T (number of scheduling days) and P (total nurses). Indices p and t represent nurses and days, respectively. Key variables A_t, B_t, and C_t denote the required nurses for morning, evening, and night shifts on the day t, forming the foundation for schedule optimisation.

Table 3 : The following notations are used to specify the model:

T	refer to the number of scheduling days
P	number of nurses available for the unit
t	is the index of days $(t = 1, ..., T)$
p	is the index of nurses $(p = 1, ..., P)$
A_p	nurse requirements for morning shift for a day t
B_p	nurse requirements for evening shift for a day t
C_p	nurse requirements for night shift for a day t

4.1.2 Decision Variable

The nurse scheduling model uses four binary decision variables: $M_{t,p}$, $E_{t,p}$, and $N_{t,p}$ to indicate if nurse p is assigned to the morning, evening, or night shift on the day t, while $C_{t,p}$ represents a day off. These variables ensure proper shift allocation.

$$M_{t,p} = \begin{cases} 1 & \text{if nurse } p \text{ is allocated to the morning shift on day } t \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

$$E_{t,p} = \begin{cases} 1 & \text{if nurse } p \text{ is allocated to the evening shift on day } t \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

$$N_{t,p} = \begin{cases} 1 & \text{if nurse } p \text{ is allocated to the night shift on day } t \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

$$C_{t,p} = \begin{cases} 1 & \text{if nurse } p \text{ is assigned a day off on day } t \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

4.1.3 Deviation Variable

Table 4 defines deviation variables that measure unmet scheduling preferences. These variables support soft constraints related to fatigue prevention, fair workload distribution, and adequate rest. Each represents underachievement or overachievement of a condition, helping to refine the schedule.

Table 4 : The deviation that has been considered for this model

d_1^-	underachievement of a nurse allocated morning shift followed by an evening or night shift on the next day.
d_1^+	overachievement of a nurse allocated morning shift followed by an evening or night shift on the next day
d_2^-	underachievement of a nurse allocated evening shift followed by a night or morning shift on the next day
d_2^+	overachievement of a nurse allocated evening shift followed by a night or morning shift on the next day
d_3^-	underachievement of a nurse allocated night shift followed by a morning or evening shift on the next day
d_3^+	overachievement of a nurse allocated night shift followed by a morning or evening shift on the next day
d_4^-	underachievement of a nurse has at least one day off.
d_4^+	overachievement of a nurse has at least one day off.

4.1.4 Model

Objective Function
The objective function of the model is to minimise shift transitions, ensure nurses have equal rest days, and maintain a balanced workload among all nurses. This is achieved by calculating the positive and negative deviations of the soft constraints. Each goal of the soft constraint will be minimised to achieve the best result. Goal 1 aims to minimise the positive deviation to avoid a morning shift followed by an evening or night shift. Goal 2 also aims to minimise positive deviation in order to avoid an evening shift followed by a morning or night shift. Goal 3 also aims to minimise positive deviation in order to avoid a night shift followed by a morning or evening shift. Hence, goal 4 aims to minimise the negative deviation in order to guarantee at least one day off for each nurse. Thus, the objective function for the model is as below:

[Minimising Total Deviation]

$$\sum_t \sum_p d_1^+ + \sum_t \sum_p d_2^+ + \sum_t \sum_p d_3^+ + \sum_p d_4^-, \tag{11}$$

[Subject to]

Hard Constraint

The scheduling model requires at least five nurses for morning and evening shifts, while exactly three nurses must be assigned to the night shift.

$$\sum_{p=1}^{P} M_{t,p} \geq 5, \quad t = 1, 2, \ldots, T \tag{12}$$

$$\sum_{p=1}^{P} E_{t,p} \geq 5, \quad t = 1, 2, \ldots, T \tag{13}$$

$$\sum_{p=1}^{P} N_{t,p} = 3, \quad t = 1, 2, \ldots, T \tag{14}$$

Only one shift per day is required of each nurse.

$$M_{t,p} + E_{t,p} + N_{t,p} + C_{t,p} = 1, \quad t = 1, 2, \ldots, T \text{ and } p = 1, 2, \ldots, P. \tag{15}$$

Each nurse works according to the number of required shifts per schedule.

$$\sum_{t=1}^{T} M_{t,p} \geq A_p, \quad p = 1, 2, \ldots, P \tag{16}$$

$$\sum_{t=1}^{T} E_{t,p} \geq B_p, \quad p = 1, 2, \ldots, P \tag{17}$$

$$\sum_{t=1}^{T} N_{t,p} \geq C_p, \quad p = 1, 2, \ldots, P \tag{18}$$

Soft Constraints

A morning shift should not be followed by an evening or night shift for the following day:

$$M_{t,p} + E_{t+1,p} + N_{t+1,p} \leq 1, t = 1, 2, \ldots, T \text{ and } p = 1, 2, \ldots, P \tag{19}$$

$$M_{t,p} + E_{t+1,p} + N_{t+1,p} + d_1^- - d_1^+ \leq 1, t = 1, 2, \ldots, T-1 \text{ and } p = 1, 2, \ldots, P \tag{20}$$

A evening shift should not be followed by an morning or night shift for the following day:

$$E_{t,p} + M_{t+1,p} + N_{t+1,p} \leq 1, t = 1, 2, \ldots, T \text{ and } p = 1, 2, \ldots, P \tag{21}$$

$$E_{t,p} + M_{t+1,p} + N_{t+1,p} + d_2^- - d_2^+ \leq 1, t = 1, 2, \ldots, T-1 \text{ and } p = 1, 2, \ldots, P \tag{22}$$

A night shift should not be followed by an morning or evening shift for the following day:

$$N_{t,p} + M_{t+1,p} + E_{t+1,p} \leq 1, t = 1, 2, \dots, T \text{ and } p = 1, 2, \dots, P \quad (23)$$

$$N_{t,p} + M_{t+1,p} + E_{t+1,p} + d_3^- - d_3^+ \leq 1, t = 1, 2, \dots, T-1 \text{ and } p = 1, 2, \dots, P \quad (24)$$

Each nurse has at least one day off per week.

$$O_{p,t} + O_{p,t+1} \leq 1, t = 1, 2, \dots, T-1 \text{ and } p = 1, 2, \dots, P \quad (25)$$

$$O_{p,t} + O_{p,t+1} + d_4^- - d_4^+ \leq 1, t = 1, 2, \dots, T-1 \text{ and } p = 1, 2, \dots, P \quad (26)$$

4.1.5 Algorithm Performance

Table 5 : Simple Data Testing

No. Nurse	No. Nurse of Morning Shift	No. Nurse of Evening Shift	No. Nurse of Night Shift	No. of Days
5	2	2	1	2

The simplified nurse scheduling scenario used in this test was designed to evaluate the basic performance of the BA under a controlled setup. The dataset included two days of scheduling, three shifts per day (morning, evening, and night), and five available nurses. The primary constraint was that each nurse could be assigned to only one shift per day, ensuring fairness and feasibility.

To assess the algorithm's convergence behaviour and determine a suitable stopping criterion, we experimented with varying numbers of iterations: 5, 10, 25, 50, and 100. The purpose of these tests was to observe how the objective function evolves as the algorithm explores the solution space over time.

As shown in Figure 3, the objective function values fluctuate initially but tend to stabilise and improve with more iterations. At lower iteration counts (e.g., 5 or 10), the algorithm lacks sufficient opportunity to thoroughly explore and exploit the solution space, resulting in suboptimal scheduling outcomes. However, as the number of iterations increases, particularly beyond 25, the algorithm begins to produce better solutions that minimise the objective function.

By 100 iterations, the improvement in the objective function indicates that the BA has effectively learned and refined a more optimal schedule, validating the effectiveness of using an iterative optimisation approach.

These findings demonstrate the importance of tuning the number of iterations: too few may result in premature convergence. In contrast, sufficient iterations enable the algorithm to strike a balance

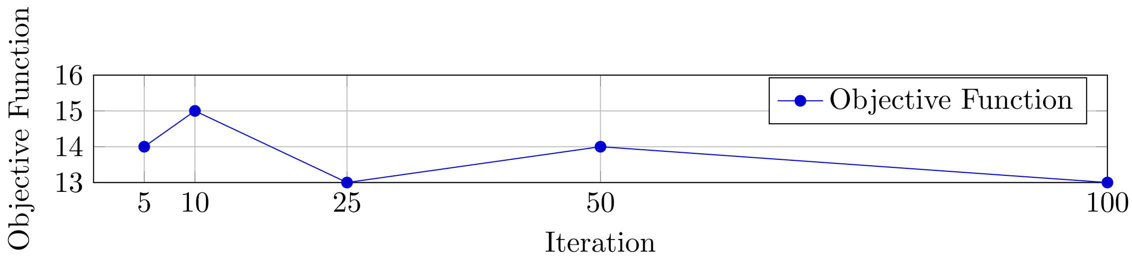

Figure 3 : Objective function values across iterations for 2-day scheduling.

Table 6 : Data from [7] Testing

No. Nurse	No. Nurse of Morning Shift	No. Nurse of Evening Shift	No. Nurse of Night Shift	No. of Days
15	5	5	3	7

between exploration and exploitation, resulting in improved scheduling performance.

The 7-day scheduling scenario involved 15 nurses with three daily shifts, requiring five nurses for the morning and evening shifts and 3 for the night shift. This setup reflects a more realistic and complex scheduling environment.

As shown in Figure 4, the objective function values improve significantly within the first 100 iterations, with only minor enhancements observed beyond that point. This confirms that the BA effectively identifies better solutions early in the optimisation process, and additional iterations help fine-tune the results.

Given these observations, 400 iterations were chosen for this experiment to ensure sufficient exploration of the solution space. This approach helps maintain solution quality while satisfying all constraints and promoting a fair and balanced nurse schedule over the 7-day period.

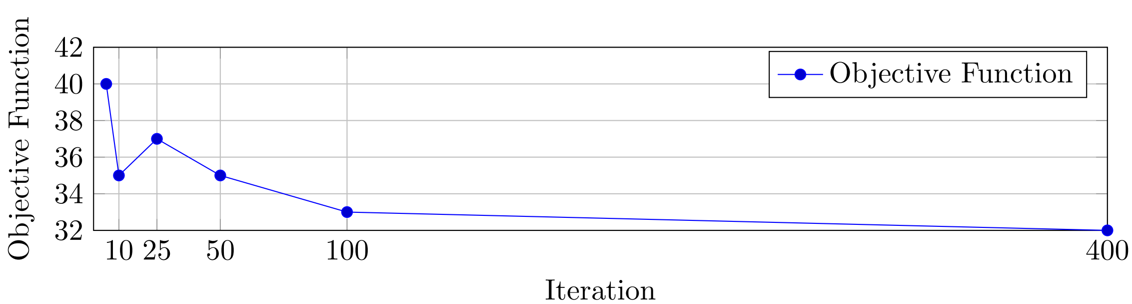

Figure 4 : Objective function values for 7-day scheduling.

5 RESULTS AND DISCUSSION

The nurse schedule in Table 8 was generated using the BA with the objective of minimising soft constraint violations. Compared to GP, shown in Table 7, which produced an objective function value of 47, the BA achieved a significantly lower value of 32. This reduction indicates better satisfaction with schedule constraints and more efficient shift allocations.

The BA effectively minimised undesirable shift transitions, particularly from morning to evening/night shifts and from evening to morning/night shifts. It also ensured that each nurse received at least one rest day during the scheduling period, fully satisfying this critical soft constraint. Night shifts were more evenly distributed among nurses, contributing to reduced fatigue and promoting fairness. In terms of computational performance, the BA completed the scheduling process in an average of 7.67 seconds using a standard computing environment (AMD Ryzen 7, 16 GB RAM). This achievement demonstrates its practical efficiency and suitability for hospital environments where scheduling decisions must be made quickly and reliably.

Overall, the BA demonstrated better performance compared to GP, not only in achieving a lower objective function value but also in producing a schedule that enhances nurse well-being and operational efficiency. Future improvements could involve parameter tuning to further refine the results and exploring adaptive strategies to improve convergence behaviours. These enhancements may lead to even more efficient and scalable scheduling solutions for broader healthcare applications.

Table 7 : 7 Days Nurse Schedule for Goal Programming

Shift	**Day 1**	**Day 2**	**Day 3**	**Day 4**	**Day 5**	**Day 6**	**Day 7**
Morning	Nurse 2	Nurse 2	Nurse 1	Nurse 5	Nurse 4	Nurse 3	Nurse 1
	Nurse 4	Nurse 5	Nurse 2	Nurse 7	Nurse 5	Nurse 5	Nurse 2
	Nurse 5	Nurse 7	Nurse 6	Nurse 8	Nurse 6	Nurse 6	Nurse 3
	Nurse 6	Nurse 8	Nurse 13	Nurse 10	Nurse 8	Nurse 12	Nurse 11
	Nurse 7	Nurse 11	Nurse 14	Nurse 14	Nurse 12	Nurse 13	Nurse 15
Evening	Nurse 1	Nurse 1	Nurse 3	Nurse 1	Nurse 3	Nurse 1	Nurse 4
	Nurse 8	Nurse 6	Nurse 8	Nurse 2	Nurse 7	Nurse 4	Nurse 5
	Nurse 9	Nurse 12	Nurse 10	Nurse 4	Nurse 11	Nurse 7	Nurse 6
	Nurse 12	Nurse 14	Nurse 12	Nurse 9	Nurse 14	Nurse 14	Nurse 10
	Nurse 13	Nurse 15	Nurse 15	Nurse 15	Nurse 15	Nurse 15	Nurse 13
Night	Nurse 3	Nurse 3	Nurse 4	Nurse 3	Nurse 1	Nurse 2	Nurse 8
	Nurse 10	Nurse 4	Nurse 9	Nurse 6	Nurse 10	Nurse 8	Nurse 9
	Nurse 11	Nurse 13	Nurse 11	Nurse 12	Nurse 13	Nurse 9	Nurse 12

Table 8 : 7 Days Nurse Schedule for Bat Algorithm

Shift	**Day 1**	**Day 2**	**Day 3**	**Day 4**	**Day 5**	**Day 6**	**Day 7**
Morning	Nurse 3	Nurse 3	Nurse 1	Nurse 2	Nurse 1	Nurse 4	Nurse 2
	Nurse 9	Nurse 9	Nurse 2	Nurse 3	Nurse 5	Nurse 6	Nurse 4
	Nurse 10	Nurse 10	Nurse 3	Nurse 5	Nurse 6	Nurse 9	Nurse 5
	Nurse 12	Nurse 11	Nurse 10	Nurse 6	Nurse 8	Nurse 13	Nurse 14
	Nurse 14	Nurse 12	Nurse 14	Nurse 14	Nurse 13	Nurse 14	Nurse 15
Evening	Nurse 1	Nurse 1	Nurse 4	Nurse 4	Nurse 2	Nurse 2	Nurse 3
	Nurse 4	Nurse 4	Nurse 7	Nurse 8	Nurse 7	Nurse 5	Nurse 10
	Nurse 8	Nurse 7	Nurse 11	Nurse 9	Nurse 10	Nurse 7	Nurse 11
	Nurse 11	Nurse 8	Nurse 13	Nurse 10	Nurse 11	Nurse 11	Nurse 12
	Nurse 15	Nurse 13	Nurse 15	Nurse 13	Nurse 12	Nurse 12	Nurse 13
Night	Nurse 5	Nurse 2	Nurse 6	Nurse 1	Nurse 3	Nurse 1	Nurse 1
	Nurse 6	Nurse 5	Nurse 9	Nurse 7	Nurse 9	Nurse 8	Nurse 7
	Nurse 7	Nurse 6	Nurse 12	Nurse 15	Nurse 15	Nurse 15	Nurse 8

6 CONCLUSION

This study demonstrated the effectiveness of the BA in optimising nurse scheduling in the emergency department. By focusing on workload balance and minimising undesirable shift transitions, the algorithm successfully generates fair and efficient schedules while adhering to both hard and soft constraints. Compared to GP, the BA showed better performance, achieving a lower objective function value (32 vs. 47) and reducing violations related to shift transitions and rest periods.

One of the key strengths of the BA is its ability to distribute workloads more equitably while ensuring that each nurse receives adequate rest, including at least one day off per week. By preventing consecutive shifts that could lead to excessive fatigue, the algorithm contributed to both operational efficiency and nurse well-being. These findings highlight the potential of BA as a practical and effective approach for nurse scheduling, offering a balance between optimisation and real-world application.

Future research could explore the implementation of adaptive stopping criteria, such as terminating the algorithm when improvements fall below a specified threshold over consecutive iterations. This could enhance computational efficiency by reducing unnecessary iterations while still ensuring high-quality solutions. Additionally, comparing BA with other heuristic and metaheuristic methods, such as Tabu Search or Simulated Annealing, could offer a deeper understanding of its relative strengths and limitations. Testing the algorithm on larger datasets or in various hospital environments would also offer valuable insights into its scalability and adaptability for real-world applications.

ACKNOWLEDGEMENT

This research was supported by the valuable contributions and encouragement of several individuals and institutions. First and foremost, I would like to express my sincere gratitude to my parents, Mohd Haizani bin Mahmod and Hasmah binti Abdullah, for their unwavering love, support, and motivation throughout this academic journey.

Special appreciation is extended to my supervisor, Madam Nor Aliza Abd Rahmin, for her dedicated guidance, insightful feedback, and continuous support, which were essential to the success of this research. I would also like to thank my second reader, Dr. Risman Mat Hasim, for his valuable input and constructive suggestions that helped refine this work. I also wish to acknowledge the support of my friends, whose encouragement and companionship provided strength during the challenging phases of this project.

Finally, I extend my gratitude to all individuals and parties who contributed, directly or indirectly, to the completion of this research.

REFERENCES

[1] Z. A. Abdalkareem, A. Amir, M. A. Al-Betar, P. Ekhan, and A. I. Hammouri, "Healthcare scheduling in optimization context: a review," *Health and Technology*, vol. 11, pp. 445–469, 2021.

[2] M. Mohammadian, M. Babaei, M. A. Jarrahi, and E. Anjomrouz, "Scheduling nurse shifts using goal programming based on nurse preferences: a case study in an emergency department," *International Journal of Engineering*, vol. 32, no. 7, pp. 954–963, 2019.

[3] A. Cinar, F. S. Salman, and B. Bozkaya, "Prioritized single nurse routing and scheduling for home healthcare services," *European Journal of Operational Research*, vol. 289, no. 3, pp. 867–878, 2021.

[4] J.-J. Wang, Z. Dai, A.-C. Chang, and J. J. Shi, "Surgical scheduling by fuzzy model considering inpatient beds shortage under uncertain surgery durations," *Annals of Operations Research*, vol. 315, no. 1, pp. 463–505, 2022.

[5] J. Mason, S. Secord, and D. MacDougall, "Interventions intended to alleviate emergency department overcrowding," *Canadian Journal of Health Technologies*, vol. 3, no. 10, 2023.

[6] Z. Fourati, S. Smaoui, and H. Kamoun, "An integrated lexicographic goal programming and dynamic satisfaction function model for effective nurse scheduling," *Decision Analytics Journal*, vol. 9, p. 100349, 2023.

[7] T. N. T. Farid and H. S. Pheng, "Nurse scheduling in hospital emergency department using goal programming," 2023.

[8] S. Patil, V. Jain, and Y. Mei, "Developing optimal nurse schedule for improving healthcare operations in the emergency department of tertiary care indian hospital," in *26TH ANZAM CONFERENCE 2023*, 2023, p. 676.

[9] P. Rerkjirattikal, V.-N. Huynh, S. Olapiriyakul, and T. Supnithi, "A goal programming approach to nurse scheduling with individual preference satisfaction," *Mathematical Problems in Engineering*, vol. 2020, no. 1, p. 2379091, 2020.

[10] A. Goday Verdaguer, M. Kaut, A. S. Valgermo, E. K. M. Dybvik, and N. M. Pedersen, "Optimising nurse staffing at the emergency department of ålesund sykehus," 2024.

[11] E. Burke, P. De Causmaecker, and G. Vanden Berghe, "A hybrid tabu search algorithm for the nurse rostering problem," in *Simulated Evolution and Learning: Second Asia-Pacific Conference on Simulated Evolution and Learning, SEAL'98 Canberra, Australia, November 24–27, 1998 Selected Papers 2.* Springer, 1999, pp. 187–194.

[12] E. K. Burke, P. D. Causmaecker, S. Petrovic, and G. V. Berghe, "Metaheuristics for handling time interval coverage constraints in nurse scheduling," *Applied Artificial Intelligence*, vol. 20, no. 9, pp. 743–766, 2006.

[13] M. P. Hasibuan and H. Cipta, "Optimization of nursing scheduling in emergency by using genetic algorithm," *Jurnal Pijar Mipa*, vol. 19, no. 1, pp. 99–106, 2024.

[14] P.-S. Chen and Z.-Y. Zeng, "Developing two heuristic algorithms with metaheuristic algorithms to improve solutions of optimization problems with soft and hard constraints: An application to nurse rostering problems," *Applied Soft Computing*, vol. 93, p. 106336, 2020.

[15] X.-S. Yang, "Bat algorithm and cuckoo search: a tutorial," *Artificial intelligence, evolutionary computing and metaheuristics: in the footsteps of Alan turing*, pp. 421–434, 2013.