Online-Journals.org

iJOE International Journal of Online and Biomedical Engineering

iJOE | eISSN: 2626-8493 | Vol. 20 No. 14 (2024) | OPEN ACCESS

https://doi.org/10.3991/ijoe.v20i14.50571

PAPER

Intelligent System Based on Round Robin and Genetic Algorithm for Managing Nurse Schedules in Health Centres in Peru

Dana Vallejos[1], Katerin Villalobos[1], José Luis Castillo-Sequera[2], Lenis Wong[1](✉)

[1]Program Information Systems Engineering, Universidad Peruana de Ciencias Aplicadas, Lima, Perú

[2]Department of Computer Science, Universidad de Alcalá, Alcalá de Henares, Spain

pcsilewo@upc.edu.pe

ABSTRACT

The inefficient assignment of nurse shifts poses a significant problem for medical centres due to stringent constraints and preferences, such as staffing requirements and hospital needs. In this study, we address these challenges by applying an improved hybrid algorithm that combines the Round Robin (RR) and Genetic Algorithms (GA) to develop an intelligent system for managing nurse schedules in hospitals in Peru. The development of the system consists of seven phases: dataset definition, variable definition, constraint definition, applying RR algorithm, applying GA, evaluation of algorithms, and intelligent system architecture. Nurse schedule records from a medical centre in Peru were used as the dataset. A total of 53 individuals, including head nurses and nurses, participated in the proposed system. The use of the proposed system resulted in a 99.71% reduction in "execution time" and a 30.08% improvement in "fairness in shift distribution" for nurses compared to previous methods. The key findings demonstrate significant improvements in both efficiency and fairness, highlighting the potential for future applications in health centres. The unique contributions of this research lie in the enhanced hybrid algorithm and its successful implementation in a real-world medical centre in Peru.

KEYWORDS

schedule management, nurses, algorithms, artificial intelligence, genetic algorithm (GA), round robin algorithm (RR)

1 INTRODUCTION

Ensuring optimal patient care hinges on effective nurse scheduling, a critical challenge for health centres that rely on the availability and strategic assignment of nursing staff, as noted by Otero-Caicedo et al. [1] and Simic et al. [2]. Background problems include the growing demand for health services, the shortage of nurses in many areas, and the need to comply with labour regulations and quality standards

Vallejos, D., Villalobos, K., Castillo-Sequera, J.L., Wong, L. (2024). Intelligent System Based on Round Robin and Genetic Algorithm for Managing Nurse Schedules in Health Centres in Peru. *International Journal of Online and Biomedical Engineering (iJOE)*, 20(14), pp. 116–138. https://doi.org/10.3991/ijoe.v20i14.50571

Article submitted 2024-06-14. Revision uploaded 2024-08-26. Final acceptance 2024-08-27.

in medical care, as highlighted by Nobil et al. [3] and Hadwan [4]. This complex issue, known as the nurse scheduling problem (NSP), aims to achieve an optimal allocation of nurse work schedules while considering hospital needs, government regulations, and individual nurse preferences, as discussed by Leung et al. [5] and Simic et al. [6].

Globally, significant proportions of nurses in various countries, including Indonesia, Greece, and the Netherlands, express dissatisfaction with their work schedules, as reported by Rizany et al. [7]. This situation underscores the fact that these professionals often endure long shifts and heavy workloads due to inefficient schedule management, which, in turn, directly impacts their job performance, as observed by Sucapuca et al. [8]. As a result, there has been a growing interest in recent years in finding solutions to this persistent problem.

In the Peruvian context, according to Article 17 of the Labour Law for Nurses, the workweek is set at 36 hours, equivalent to 150 hours per month, as noted by Muñoz [9]. However, this profession operates in an environment where social protection policies are incomplete and the health system is not universal, which is reflected in the perception of nurses in the country. A significant percentage believe that the health system has worsened, and many express job dissatisfactions due to disorganized schedules, as highlighted by Laguna [10]. This dissatisfaction deepens further, as a large percentage of surveyed nurses disagree with their work schedules and find them uncomfortable, according to research conducted at the National Institute of Neurological Sciences by Campos et al. [11].

To address this issue, numerous studies have utilized different algorithms for schedule management, such as the Genetic Algorithm (GA), as demonstrated by Malekian et al. [12], Abayomi-Alli et al. [13], and Koruca et al. [14]. This algorithm is applied iteratively until a solution that satisfies the objectives is found. While this process can be slow, it can discover solutions that are difficult or even impossible to find using other methods. Similarly, metaheuristic and heuristic algorithms, as explored by Turhan and Bilgen [15], are designed to find satisfactory solutions to optimization problems based on imperfect or incomplete information in real-world scenarios with limited resources. Despite their effectiveness in finding optimal solutions, it is important to acknowledge that these approaches may have limitations in terms of execution speed and computational costs.

To address the challenges of nurse scheduling in hospitals, this study proposes an intelligent system that combines the Round Robin (RR) algorithm and GA for managing nurse schedules in Peruvian healthcare centres. The system leverages the RR algorithm to prioritize nurse preferences while adhering to general constraints and then optimizes the schedule using the GA. This approach aims to maximize hospital requirements' satisfaction while ensuring that nurse preferences are met to the greatest extent possible.

This paper follows a structure covering various sections. Section 2 discusses previous work related to the topic. Subsequently, Section 3 details the proposed model. Section 4 presents the experimentation. Section 5 discusses the results and discussion. Section 6 presents the expert judgment. Finally, Section 7 presents the conclusions.

2 RELATED WORK

This section reviews studies related to nurse schedule management, specifically evaluating various artificial intelligence algorithms, including GAs as noted by Malekian et al. [12], Hybrid Algorithms by Abadi et al. [16], Heuristic Algorithms by Chen et al. [17], Hyperheuristics by Kheiri et al. [18], Metaheuristics by Turhan and Bilgen [15], and Stochastic Algorithms by Legrain [19].

Chen et al. [20], combining heuristic methods such as decision trees and greedy search with metaheuristic algorithms such as the Bat Algorithm (BA) and PSO, improves the efficiency and quality of solutions for the nurse rostering problem (NRP). Zhang et al. [21] introduce MDQN-MA, a multi-agent framework that leverages metaheuristics and reinforcement learning to outperform existing algorithms in complex scheduling scenarios. Similarly, Muniyan et al. [22] present NM-ABC, a metaheuristic algorithm that enhances global search efficiency and memory management in nurse scheduling by integrating local search.

Some studies have proposed GAs to solve the NSP. In Malekian et al. [12], GA is proposed because it reduces the likelihood of getting trapped in a local minimum and often produces high-quality solutions in a shorter period. Through genetic operators such as crossover and mutation, new solutions are iteratively produced to improve the quality of assignments, as noted by Abayomi-Alli et al. [13] and Koruca et al. [14]. Some hybrid approaches combine the GA with other methods to address specific aspects of the problem, such as nurse fatigue, as discussed by Amindoust et al. [23]. Additionally, Tabu search properties are incorporated to reduce complexity and execution time, as highlighted by Abayomi-Alli et al. [13]. Parameters are adjusted to balance exploration and exploitation of the search space, according to Koruca et al. [14]. This process continues until an optimal or near-optimal solution that satisfies the problem's demands and constraints is found.

Several studies have adopted different hybrid algorithms to solve the NSP more efficiently. These include combinations of salp swarm and GAs, as discussed by Abadi et al. [16], heuristic and metaheuristic algorithms, also by Abadi et al. [16], MCTS and HC, as explored by Goh et al. [24], MOGWO and heuristic algorithms, according to Huang et al. [25], and two hybrid approaches combining the harmony search algorithm (HSA) and the artificial immune system (AIS), as noted by Hadwan et al. [4]. Additionally, hospital regulations were considered, as highlighted by Chen et al. [26]. In the first stage, the GA was found suitable for solving nurse scheduling without violating any established regulations. In the second stage, solutions obtained using GA were combined with nurse preferences, as noted by Abadi et al. [16] and Chen et al. [26].

Various studies have proposed heuristic algorithms to solve personnel scheduling and resource allocation problems in hospital settings. Some approaches combine predefined heuristics, as noted by Huang et al. [25], and deep learning, as discussed by Chen et al. [17], to generate quality solutions for nurse scheduling. Additionally, the complementary use of heuristics and metaheuristics has been employed to satisfy constraints and improve solution quality, as highlighted by Abadi et al. [16]. Other investigations have integrated heuristics within more complex hybrid algorithms, combining them with multi-objective optimizers, as explored by Huang et al. [25], search techniques such as Hill Climbing and Iterated Local Search, according to Goh et al. [24], or deep learning models trained with previous instances, as indicated by Chen et al. [25]. In certain cases, the goal is to generate feasible solutions quickly, as emphasized by Abadi et al. [16], while in others, the aim is to explore the solution space to escape local optima, as shown by Goh et al. [24] and Chen et al. [26]. Heuristics have also been applied to assign medical staff among hospitals based on criteria such as size and penalties, as demonstrated by Chen et al. [26], allowing for efficient approximate solutions when calculating an optimal assignment is computationally costly.

In Kheiri et al. [18], a metaheuristic approach is introduced that integrates PSO and IP to create optimized schedules. Unlike traditional evolutionary algorithms, the PSO algorithm avoids using crossover and mutation to generate new solutions, instead relying on particle velocities to develop new configurations. Similarly, in Turhan and Bilgen [15], the NRP is addressed by employing the F&R heuristic as a foundation to derive high-quality initial solutions. The NRP is divided according to the number of nurses and the duration in weeks. Initially, the F&R heuristic is applied to obtain a solution, which is then refined within the Simulated Annealing (SA) framework. Various neighbourhood structures are explored in subsequent iterations to further enhance the initial solution.

Other algorithms have been applied to address NSPs, including hyperheuristics and stochastic algorithms. The sequence-based hyperheuristic (SSHH) proposed by Turhan and Bilgen [27] utilizes a hidden Markov model to learn successful transitions between low-level heuristics, thereby generating an effective sequence of heuristics. Finally, Legrain et al. [19] apply a stochastic algorithm that combines a primal-dual approach with sample average approximation (SAA), utilizing linear programming and branch-and-cut and price (BCP) frameworks.

3 MATERIALS AND METHODS

This section presents a hybrid Round Robin and Genetic Algorithm (RRGA) to manage nurse schedules in healthcare facilities in Peru. The RR method is used to generate an initial schedule that satisfies the fundamental constraints, which is then refined using the GA to address the most stringent constraints and preferences. This methodology encompasses seven phases: (1) definition of the dataset, (2) definition of variables, (3) definition of constraints, (4) applying RR algorithm, (5) applying GA, (6) evaluation of algorithms, and (7) intelligent system architecture.

3.1 Dataset definition

For the development of the proposed algorithm, the dataset provided by the Virgen del Carmen Maternal and Child Health Centre, containing 45 records of nurse shift preferences, was used. These data underwent a cleaning and transformation process to ensure compatibility with the algorithm's requirements.

Data cleaning and transformation. Data cleaning was performed by removing records of nurses on vacation, as they did not require shift assignments. Shift preferences were then encoded into numerical values to facilitate algorithmic handling as shown in Table 1.

Data tabulation. The data structuring was carried out using three separate tables to facilitate interpretation and subsequent processing. Table 2 contains the shift preferences of the nurses for one week. Table 3 shows the days off selected by each nurse according to the initial schedule established. Finally, the system considers the number of nurses required per shift. Thus, five nurses are assigned to the morning shift, four to the evening shift, and three to the night shift.

Table 1. Data transformation from the Virgen del Carmen Maternal and Child Health Centre

Variables	Initial Value	Transformation
Day Shift	D	1
Afternoon Shift	A	2
Night Shift	N	3
Days Off	–	0

Table 2. Nurse shift preferences

Nurses	Weekdays						
	1	2	3	4	5	6	7
1	3	1	0	3	2	2	2
2	2	3	1	3	0	1	3
3	1	3	2	0	2	1	1
4	3	3	1	2	1	0	3
5	3	3	1	0	3	2	2

Table 3. Day off preferences

Nurses	Day off
1	3
2	5
3	4
4	6
5	4

3.2 Definition of variables

The scheduling of nurses requires considering a series of parameters to generate work shifts that adequately cover the needs of both the nurses and the hospital or clinic. Table 4 shows the variables identified according to [14], which are used to properly model scheduling constraints, coverage, rest periods, and preferences necessary to generate feasible and balanced schedules for the nurses.

Table 4. Variables for nurse schedule programming

Id	Variables	Description
V01	Total nurses	Number of nurses using the system
V02	Shifts	Number of shifts nurses can choose per day
V03	Number of nurses per shift	Number of nurses required per shift
V04	Shift preferences	Shift preferences chosen by each nurse for one week
V05	Days off preferences	Day of the week nurses choose as their day off

3.3 Definition of constraints

Constraints in NSP are typically categorized into two main types: hard constraints and soft constraints, as identified in several prior studies on the topic. The primary objective in managing nurse schedules is to maximize the utilization of human resources by developing a schedule that fully satisfies all hard constraints while accommodating soft constraints to the greatest extent possible. By classifying the most common constraints as either hard or soft based on their level of importance, the aim is to generate feasible shift schedules that optimize nurse well-being while adhering to essential hospital requirements.

Hard constraints (Mandatory): Hard constraints are limitations that must be met in any feasible solution to the problem. Table 5 provides a list of the identified hard constraints based on the studies [13] [14] [21].

Table 5. Hard constraints

Id	Constraints
HC1	No nurse can work the night shift and then directly switch to the morning shift.
HC2	Nurses must have at least 1 day off per week.
HC3	Nurses can work a maximum of 6 shifts per week.
HC4	A nurse cannot cover more than 1 shift per day.

These constraints will also serve as metrics to evaluate the RRGA algorithm's performance. They are represented in equations (1), (2), (3), and (4).

$$HC1 = \left(1 - \frac{\sum_{d=1}^{D-1} \sum_{j=1}^{N} 1\left(j \in A_{d,S} \wedge j \in A_{d+1,1} \right)}{N \cdot (D-1)} \right) \times 100 \tag{1}$$

$$HC2 = \left(1 - \frac{\sum_{j=1}^{N} 1\left(WorkedDays_j \geq 6 \right)}{N} \right) \times 100 \tag{2}$$

$$HC3 = \left(\frac{\sum_{j=1}^{N} 1(ShiftsWorked_j > 6)}{N} \right) \times 100 \tag{3}$$

$$HC4 = \left(\frac{\sum_{d=1}^{D} \sum_{j=1}^{N} 1(MoreThanOneShift_j(d))}{N \cdot D} \right) \times 100 \tag{4}$$

Where

- N = Total number of nurses
- D = Total number of days in the schedule
- S = Total number of shifts per day
- A_{ds} = Set of nurses assigned to shift s on day d.
- j = Identifier of a specific nurse (varies from 1 to N)
- d = Represents a specific day in the schedule (varies from 1 to D)
- *1* = Indicator function (true)

Soft constraints (Optional). Unlike hard constraints, soft constraints are not mandatory but represent desirable criteria to improve the quality of nurse schedules. Table 6 lists the identified soft constraints based on the studies [13] [14] [21].

Table 6. Soft constraints

Id	Constraints
SC1	A nurse can have up to 2-night shifts per week.
SC2	Grant requested days off for nurses as much as possible.
SC3	Assign requested shifts for nurses as much as possible.

3.4 Applying round robin algorithm

The RR algorithm is developed in Python to assign shifts to nurses. The algorithm begins by reading the data from an Excel file and transforming it into suitable data structures (Step 1). Then, it assigns days off to the nurses according to the provided data (Step 2). Next, it assigns shifts using the RR method, prioritizing the nurses' preferences and checking their availability (Steps 3 and 4). If there are unassigned shifts, they are filled with available nurses (Step 5). Subsequently, the assignment is checked and adjusted to ensure that there are no unassigned shifts and to avoid assigning a shift on a nurse's day off (Step 6). Finally, the final assignment of shifts and days off is presented in dictionaries (Step 7). The generated outputs will be used as input data for the genetic algorithm.

The metrics used to evaluate the RR algorithm were collected from [13]. These are the percentage of preferred free shifts assigned and the percentage of organized nurses, represented by Eqs. (5) and (6), respectively. Additionally, computational time (CT) is considered.

$$\%\,of\ PFSA = \left(\frac{No.\,of\ PFSA}{TA}\right) \times 100 \tag{5}$$

$$\%\,of\ ON = \left(\frac{No.\,of\ ON}{TN}\right) \times 100 \tag{6}$$

Where,

- *PFSA*: Preferred free shifts assigned
- *TA*: Total shifts assigned
- *TN*: Total nurses
- *ON*: Organized nurses

3.5 Applying genetic algorithm

The GA, developed in Python, follows the process shown in Figure 1. It starts by receiving a preliminary assignment from the RR algorithm, where an initial population of potential schedules is created, complying with the constraints SC1 and SC3 (Step 1). Each schedule is evaluated based on its 'fitness', considering the preferences for days off and complying with constraint SC2 (Step 2). Natural selection is simulated by choosing schedules with higher 'fitness' for reproduction, ensuring genetic diversity (Step 3). This is followed by crossover between selected schedules at random points to produce offspring (Step 4). Mutations are introduced randomly to explore new solutions (Step 5), and if the constraints HC1, HC2, HC3, and HC4 are violated, they are corrected in a chromosome repair process (Step 6). Subsequently, the population is renewed with the new candidates (Step 7), and this process iterates over several generations (Step 8). Finally, the best individual is chosen (Step 9), the one with the highest 'fitness' and best meeting the established needs and preferences, and the final schedule that meets the established constraints is provided (Step 10).

The metrics used to evaluate the GA were collected from [13]. These are Eqs. (5) and (6). Additionally, the fitness function, Eq. (7), is considered as part of the metrics according to [13] [14] [15], as well as the computational time (CT).

$$\sum_{j=1}^{N}\sum_{d=1}^{D}\left[\left(d=P_j\right)\cdot\left(1-\sum_{s=1}^{S}1\left(j\in A_{ds}\right)\right)\right] \tag{7}$$

Where,

- N = Total number of nurses
- D = Total number of days in the schedule
- S = Total number of shifts per day
- j = Identifier of a specific nurse (varies from 1 to N)
- d = Represents a specific day in the schedule (varies from 1 to D).
- P_j = Preferred day off for nurse j.
- A_{ds} = Set of nurses assigned to shift s on day d.

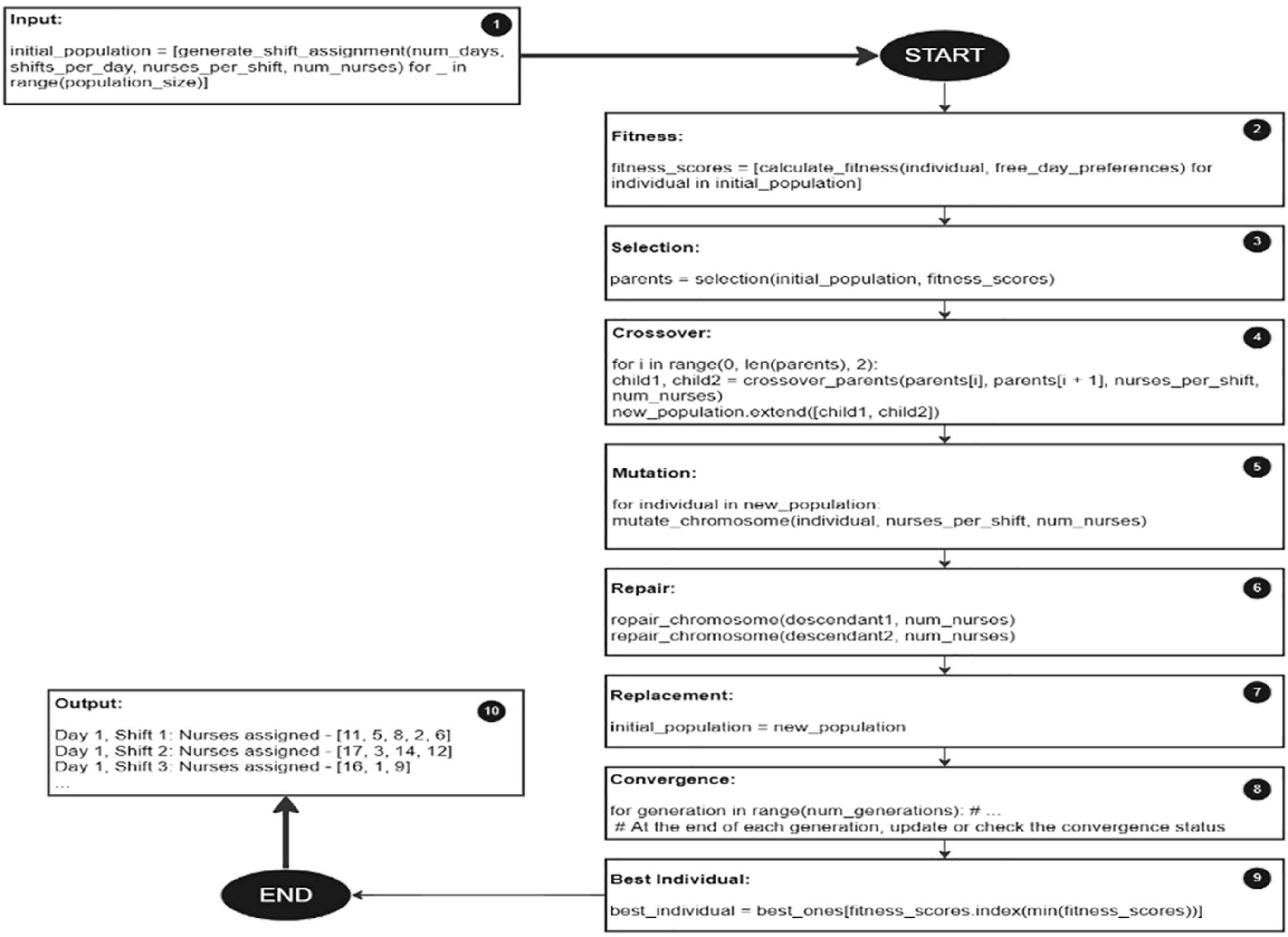

Fig. 1. Genetic algorithm in Python

3.6 Evaluation of algorithms

In this section, the hybrid RRGA algorithm is evaluated using Google Colab and a dataset provided by the triage area of the Virgen Del Carmen Maternal and Child Health Centre. Twenty scenarios were examined, including a total of 591 nurse records, representing the shift preferences of a group ranging from 15 to 40 nurses over one week, covering three daily shifts (morning, afternoon, and night) and the demand for nurses per shift. Table 7 details these cases, showing the total number of nurses (N) and the number required for each shift: morning (NMS), afternoon (NAS), and night (NNS).

Table 7. Results of the RRGA

Case	N	NMS	NAS	NNS
1	29	12	9	3
2	32	12	9	3
3	19	7	3	2
4	36	12	10	5
5	39	10	8	3
6	28	9	5	3
7	25	10	7	2
8	25	8	6	2
9	17	6	3	2
10	15	5	4	2
11	40	16	8	4
12	40	13	12	7
13	35	15	9	3
14	21	7	5	3
15	35	14	8	3
16	35	12	9	4
17	34	13	8	3
18	30	9	7	2
19	28	10	6	3
20	28	11	6	4

Figure 2 shows six 'fitness' graphs corresponding to the first six cases from Table 7, plotting the progression of the RRGA over the first 50 generations: case 1 (Figure 2a), case 2 (Figure 2b), case 3 (Figure 2c), case 4 (Figure 2d), case 5 (Figure 2e), and case 6 (Figure 2f). Each point in the graphs represents the 'fitness' value of the best chromosome of its respective generation. The decreasing trend indicates a continuous improvement in the assignment of preferred free shifts through the evolutionary process.

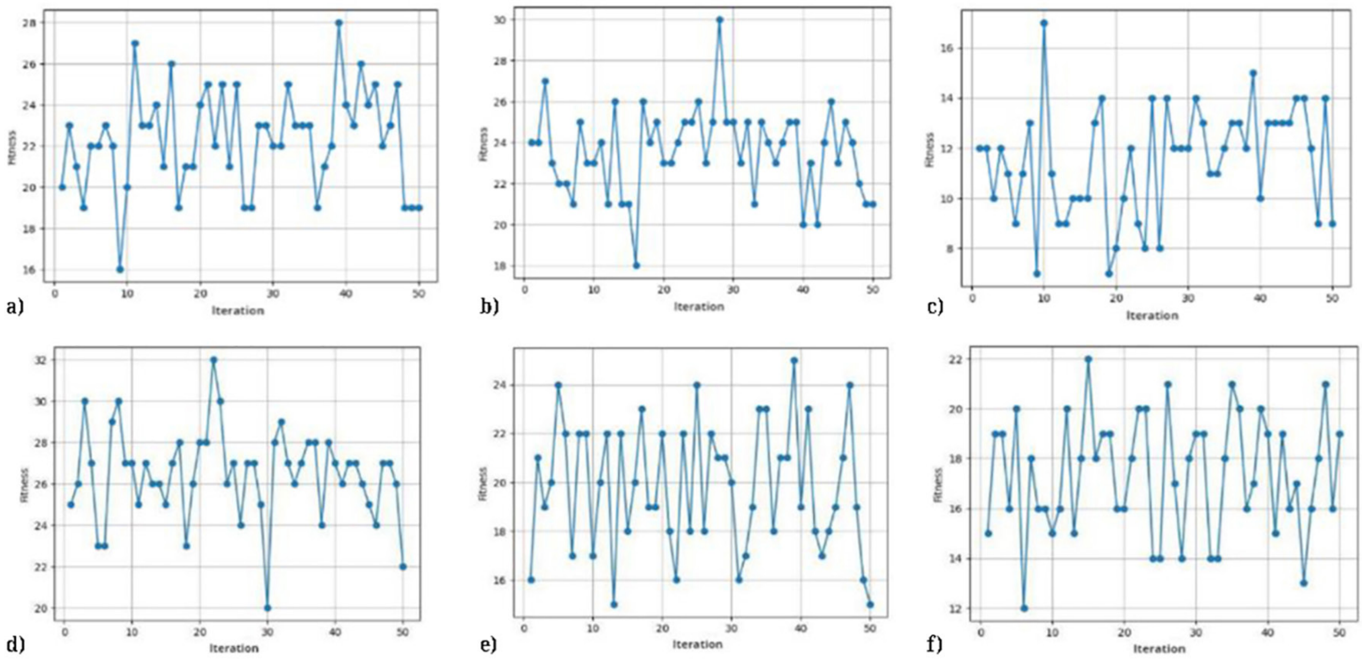

Fig. 2. Fitness of the round robin and genetic algorithm in cases 1 to 6

Table 8 shows the performance of the RR, GA, and RRGA algorithms, highlighting that all achieved 100% in the organization of the nurses. The RR excels by ensuring 100% assignment of preferred free shifts in each case, while the GA and RRGA vary, with 22.86% to 56.66% and 17.5% to 52.63%, respectively. In terms of speed, the RR stood out with 0.1 seconds of computational time, while the GA and RRGA recorded times of four to 22 seconds and six to 23 seconds, respectively. These results indicate that although the RRGA does not surpass the RR in speed, it provides an optimal balance between time efficiency and accuracy in the assignment of free shifts.

Table 8. Comparison of metrics for the round robin, genetic algorithm, and round robin genetic algorithm

Case	RR			GA			RRGA		
	% de PFSA	% de ON	CT (s)	% de PFSA	% de ON	CT (s)	% de PFSA	% de ON	CT (s)
1	100	100	0.1	28.4	100	20	31.0	100	19
2	100	100	0.1	28.1	100	16	31.2	100	14
3	100	100	0.1	26.3	100	5	52.6	100	5
4	100	100	0.1	36.1	100	22	27.7	100	20
5	100	100	0.1	33.3	100	14	43.5	100	9
6	100	100	0.1	53.5	100	12	35.7	100	8
7	100	100	0.1	40	100	11	32	100	10
8	100	100	0.1	56	100	12	40	100	9
9	100	100	0.1	35.2	100	4	47.0	100	7
10	100	100	0.1	46.6	100	9	33.3	100	6
11	100	100	0.1	37.5	100	16	37.0	100	16

(Continued)

Table 8. Comparison of metrics for the round robin, genetic algorithm, and round robin genetic algorithm *(Continued)*

Case	RR			GA			RRGA		
	% de PFSA	% de ON	CT (s)	% de PFSA	% de ON	CT (s)	% de PFSA	% de ON	CT (s)
12	100	100	0.1	35	100	24	17.5	100	23
13	100	100	0.1	22.8	100	16	28.5	100	18
14	100	100	0.1	52.3	100	11	23.8	100	7
15	100	100	0.1	28.5	100	14	40	100	14
16	100	100	0.1	31.4	100	17	36.1	100	16
17	100	100	0.1	29.4	100	14	35.2	100	14
18	100	100	0.1	56.6	100	13	43.3	100	10
19	100	100	0.1	28.5	100	10	21.4	100	9
20	100	100	0.1	42.8	100	14	32.1	100	12

Table 9 shows the comparison of hard constraints between the GA and RRGA algorithms, providing an analysis of the percentage of compliance with constraints HC1, HC2, HC3, and HC4. The results show that the GA exhibits variability in constraint compliance, with percentages ranging from 75.86% to 100% across different tests and constraints, while the RRGA achieves 100% compliance in most constraints and cases analyzed. This pattern suggests the superiority of the RRGA over the traditional GA in meeting the imposed constraints.

Table 9. Comparison of hard constraints between the Genetic Algorithm and the Round Robin algorithm

Case	GA				RRGA			
	HC1	HC2	HC3	HC4	HC1	HC2	HC3	HC4
1	95.4	75.86	75.86	100	100	93.10	93.10	100
2	98.43	84.37	84.37	100	100	100	100	100
3	97.37	100	100	100	100	100	100	100
4	97.68	86.11	86.11	100	100	100	100	100
5	98.29	100	100	100	100	100	100	100
6	98.8	96.42	96.42	100	100	100	100	100
7	98.66	84	84	100	100	100	100	100
8	98	92	92	100	100	100	100	100
9	97.06	94.12	94.12	100	100	100	100	100
10	96.66	80	80	100	100	100	100	100
11	97.50	87.50	87.50	100	100	100	100	100
12	95.83	77.50	77.50	100	100	100	100	100
13	98.09	80	80	100	100	100	100	100
14	96.82	95.23	95.23	100	100	100	100	100
15	100	82.86	82.86	100	100	100	100	100

(Continued)

Table 9. Comparison of hard constraints between the Genetic Algorithm and the Round Robin algorithm *(Continued)*

Case	GA				RRGA			
	HC1	HC2	HC3	HC4	HC1	HC2	HC3	HC4
16	98.57	97.14	97.14	100	100	100	100	100
17	97.55	82.35	82.35	100	100	100	100	100
18	97.22	96.66	96.66	100	100	100	100	100
19	97.62	96.43	96.43	100	100	100	100	100
20	97.02	89.28	89.28	100	100	100	100	100

3.7 Intelligent system architecture

The system is designed to generate nurse schedules and is tailored for two types of users: nurses and the head nurse. Figure 3 illustrates the system's physical architecture, which was developed using Node.js, Express, and TypeScript for the backend, deployed on Amazon EC2, and Flutter along with Dart for the frontend, ensuring compatibility on both Android and iOS devices. The integration of the RRGA developed in Python is achieved through Flask, deployed on AWS Lambda, to create an API that connects the backend and frontend with the PostgreSQL database managed by Amazon RDS. To generate the schedule, nurses input their weekly shift preferences, selecting preferred shifts and one day off, while the head nurse specifies the number of nurses required for each shift. Based on this data, the system automatically assigns shifts, considering both hard and soft constraints. The application is named Calenurse.

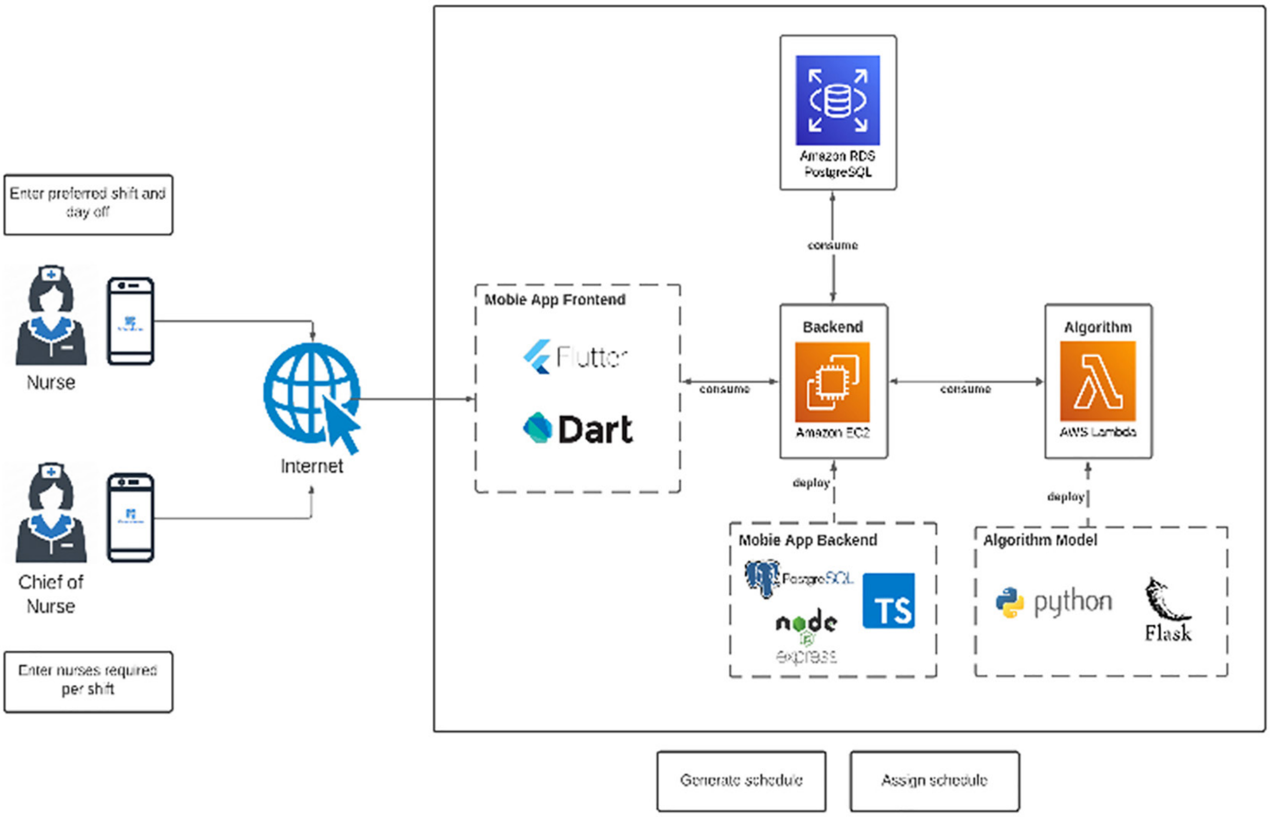

Fig. 3. Physical architecture of the system

4 EXPERIMENTATION

To validate the system, 50 nurses and 3 head nurses from the Virgen Del Carmen Maternal and Child Health Centre participated in evaluating the system's performance through two methods: (a) the manual method and (b) using a Calenurse. Both experiments were conducted over one week, as shown in Table 10.

At the end of the experiment, the following metrics were evaluated: shift coverage, equity in shift distribution, fulfilled constraints, and schedule generation time.

Table 10. Experiment scenarios

Experiment 1: Experiment Scenarios			
Group	**Participants**	**Area**	**Metrics**
Group 1	1 head nurse, 20 nurses	Emergency	Shift coverage, equity in shift distribution, fulfilled constraints, and schedule generation time
Group 2	1 head nurse, 18 nurses	ICU	
Group 3	1 head nurse, 12 nurses	Obstetrics	
Experiment 2: Using Calenurse			
Group 1	1 head nurse, 20 nurses	Emergency	Shift coverage, equity in shift distribution, fulfilled constraints, and schedule generation time
Group 2	1 head nurse, 18 nurses	ICU	
Group 3	1 head nurse, 12 nurses	Obstetrics	

Metrics:

$$SC = \frac{NSC}{SS} \times 100\% \tag{8}$$

$$EDS = \frac{\sigma}{\mu} \tag{9}$$

$$FC = \frac{NFC}{TC} \times 100\% \tag{10}$$

Where

- *SC*: Shift coverage
- *NSC*: Number of shifts covered
- *SS*: Total number of scheduled shifts
- *EDS*: Equity in the distribution of shifts
- σ: Standard deviation of the number of shifts assigned to each nurse
- μ: Average number of shifts assigned to each nurse
- *FC*: Fulfilled constraints
- *NFC*: Number of fulfilled constraints
- *TC*: Total number of constraints

4.1 Experiment 1: Manual method

The experiment focuses on the manual scheduling of nurses. The steps followed are: (i) the head nurse creates the schedule in Excel without first consulting the nurses about their preferred shifts; (ii) the schedule is then posted, and if nurses require a schedule change; (iii) they inform the head nurse and coordinate among themselves to find a substitute for the shift (see Figure 4).

4.2 Experiment 2: Using Calenurse

This experiment involves automating the process of nurse schedule generation using the Calenurse system. The procedure is shown in Figure 5 and consists of the following steps: (i) nurses choose their schedules according to their preferences; (ii) the head nurse receives these preferences and generates the schedules; (iii) subsequently, nurses have the option to submit a schedule change request if they cannot attend their assigned shifts; and (iv) finally, the head nurse reviews and decides whether to accept or reject the request.

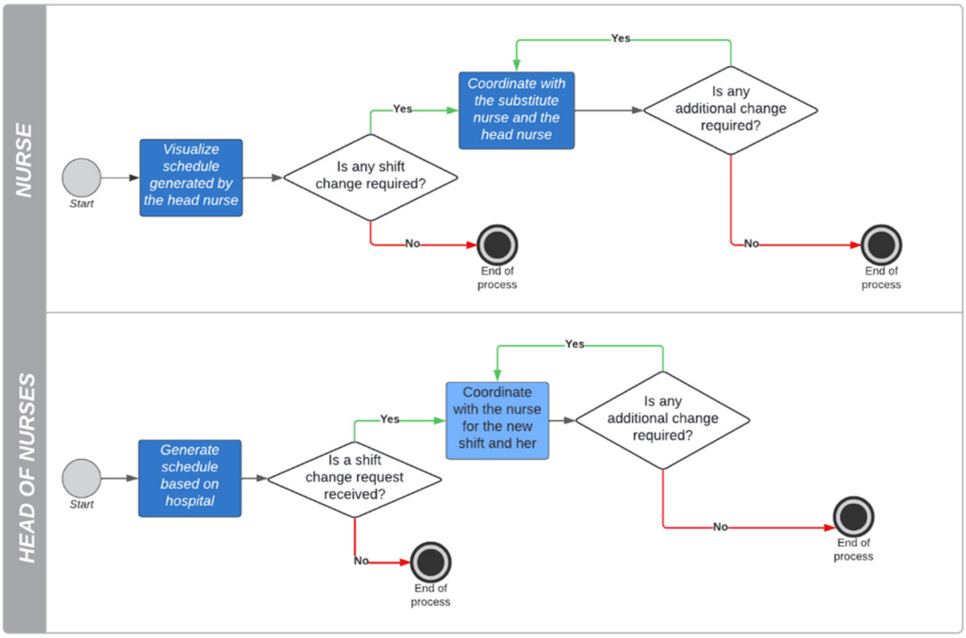

Fig. 4. Flowchart of experiment 1

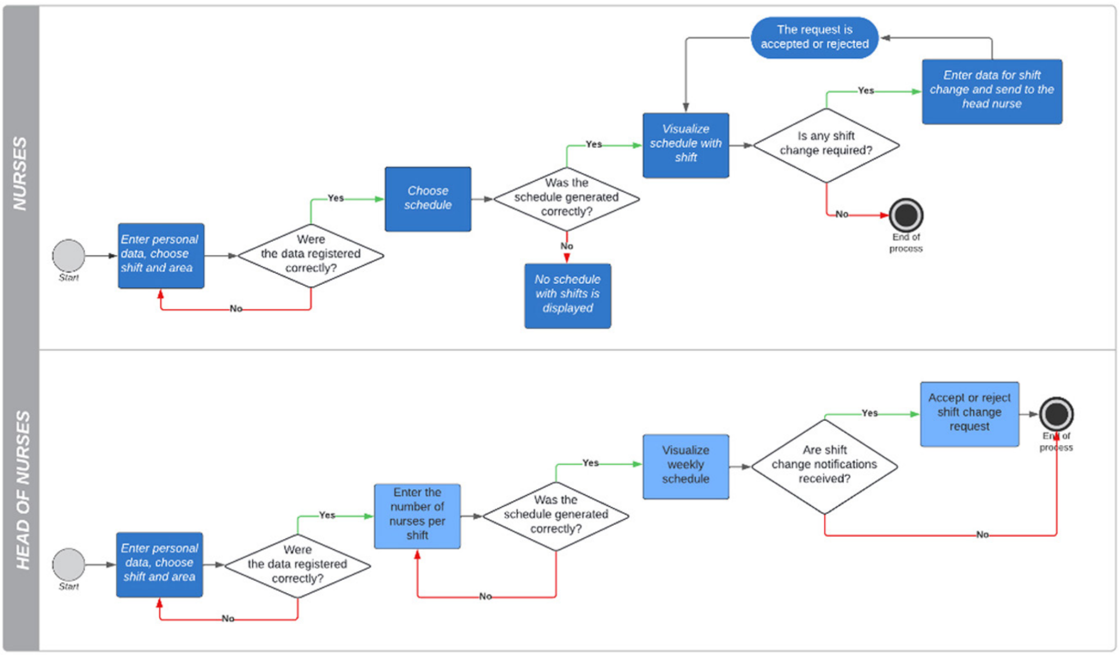

Fig. 5. Flowchart of experiment 2

4.3 Expert judgment

To understand the perceptions of head nurses and nurses regarding the quality of the system, a survey was conducted. The survey was based on quality characteristics proposed by ISO/IEC 25010, considering functionality, usability, and efficiency. Google Forms was used to create the survey, which consists of 19 closed-ended questions using the Likert scale (1 = "Strongly disagree," 2 = "Disagree," 3 = "Neutral," 4 = "Agree," 5 = "Strongly agree"). Tables 11 and 12 show the questions asked to head nurses and nurses, grouped into categories.

Table 11. Questions for head nurses

Characteristic	Question	
Functionality	Q1	Do you consider that the functionality facilitates the efficient assignment of nurses per shift?
	Q2	Do you find the functionality to accept or reject shift change requests within the system useful?
	Q3	Do you consider that the system correctly assigns nurse shifts?
	Q4	Do you think the system meets the hard constraints posed?
Usability	Q5	Do you find the system's interface intuitive for using the different functionalities?
	Q6	Does the system provide clear and useful error messages when something goes wrong?
	Q7	Does the system design facilitate the visualization and understanding of your work schedule and any changes made?
Efficiency	Q8	Do you find that the system speeds up the scheduling process compared to traditional methods?
	Q9	Do you consider that the system automates the nurse selection process per shift?
	Q10	Does the system provide you with the necessary flexibility to make quick schedule changes if needed?

Table 12. Questions for nurses

Characteristic	Question	
Functionality	Q1	Do you find that the system allows you to submit your desired schedule effectively?
	Q2	Do you find the feature of submitting schedule change requests within the system useful?
	Q3	Do you believe that the system respects nurses' preferences for schedule generation?
	Q4	Do you find that the system generates schedules that fit your personal preferences?
Usability	Q5	Do you find the system interface easy to use for selecting your schedule?
	Q6	Do you find the process of requesting a schedule change clear and straightforward?
Efficiency	Q7	Do you consider that the system streamlines the schedule assignment process compared to traditional methods?
	Q8	Do you believe that the process of selecting schedule preferences within the system helps you save time in planning your work week?
	Q9	Do you think the system helps minimize schedule changes and optimizes shift distribution efficiently?

5 RESULTS AND DISCUSSION

This section presents the results of the system, evaluating its performance in three test groups. It is observed that the number of nurses assigned to the morning shift is greater than the night shift, reflecting patient demand. Figure 6 illustrates the distribution of nurses per shift used to generate the schedule in the two experiments.

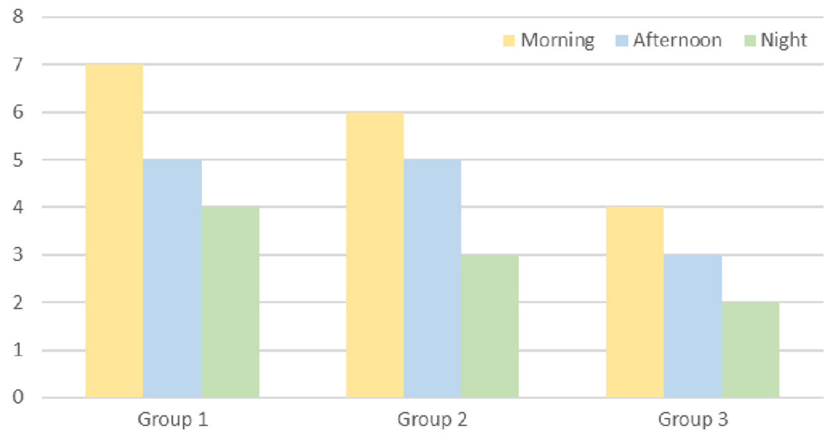

Fig. 6. Distribution of shifts by group

More discussion about the results of experiments 1 and 2 will be provided in the following subsections.

5.1 Results of experiment 1

Table 13 presents the results of Experiment 1, where the manual method was applied for schedule generation. It is noted that all groups achieved 100% shift coverage. Regarding equity in shift distribution (see Figure 7), group 1 demonstrated a higher index of 0.236 compared to groups 2 and 3. In terms of constraint compliance, group 1 achieved 90%, followed by group 2 with 87% and group 3 with 85%. Additionally, group 2 was the most efficient in schedule preparation time, with 40 minutes, followed by group 3 with 43 minutes and group 1 with 45 minutes.

Table 13. Results of experiment 1

Participants	CT	ED	RC	TG (min)
Group 1	100%	0.236	90%	45 min
Group 2	100%	0.140	87%	43 min
Group 3	100%	0.137	85%	40 min

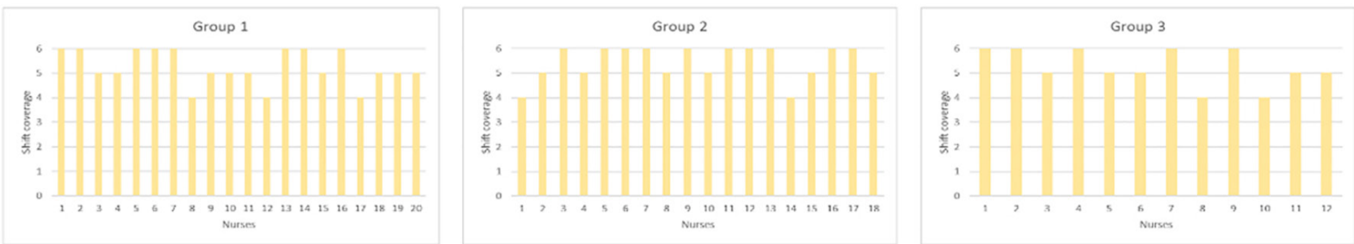

Fig. 7. Shifts distribution by nurses

5.2 Results of experiment 2

Table 14 presents the results of Experiment 2, where the Calenurse method was applied for schedule generation. It is noted that all groups achieved 100% shift coverage. Regarding equity in shift distribution (see Figure 8), a very equitable performance with low indices is observed: group 1 reached an index of 0.088, group 2 an index of 0.126, and group 3 an index of 0.113. In terms of compliance with hard constraints, all groups achieved 100%. Additionally, group 3 was the most efficient in schedule preparation time, with only 5 seconds, followed by group 2 with 8 seconds and group 1 with 9 seconds. These results highlight the effectiveness and precision of the Calenurse method, especially in reducing execution time and achieving equity in shift distribution.

Table 14. Results of experiment 2

Participants	CT	ED	RC	TG (sec)
Group 1	100%	0.088	100%	9 sec
Group 2	100%	0.126	100%	8 sec
Group 3	100%	0.113	100%	5 sec

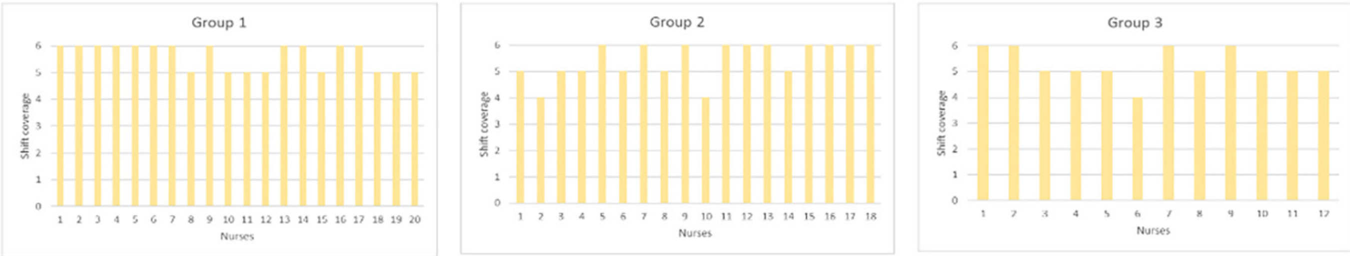

Fig. 8. Shifts distribution by nurses

5.3 Comparison between experiments 1 and 2

In Table 15, a significant improvement is observed when comparing the manual method used in Experiment 1 with the Calenurse method employed in Experiment 2. Although both experiments achieved total coverage of 100% in shift assignments, other metrics show notable improvements. Figure 9 presents a comparison of the shifts assigned to each nurse in both experiments, highlighting fairness in distribution. Experiment 2 stands out for greater fairness in shift distribution, with indices of 0.088 for Group 1, 0.126 for Group 2, and 0.113 for Group 3, compared to the higher indices of Experiment 1. Furthermore, compliance with hard constraints reached 100% in all groups of Experiment 2, significantly improving over the 90%, 87%, and 85% compliance rates of the corresponding groups in Experiment 1. Finally, the schedule generation time was drastically reduced, going from over 40 minutes per group in Experiment 1 to less than 10 seconds per group in Experiment 2, demonstrating the efficiency of the Calenurse method.

Table 15. Comparison of the results of experiments 1 and 2

Metrics	Experiment 1			Experiment 2		
	Group 1	Group 2	Group 3	Group 1	Group 2	Group 3
CT	100%	100%	100%	100%	100%	100%
ED	0.236	0.140	0.137	0.088	0.126	0.113
RC	90%	87%	85%	100%	100%	100%
TG (min/sec)	40 min	45 min	43 min	9 sec	8 sec	5 sec

Fig. 9. Comparison of experiments 1 and 2 for groups 1, 2, and 3

6 EXPERT JUDGMENT

Figure 10 presents the results of the survey conducted with three nursing supervisors (N), evaluating three categories: functionality, usability, and efficiency, through

10 questions. 100% of the experts rated questions Q1, Q3, Q4, Q5, Q6, Q8, Q9, and Q10 with a score of 5 (very satisfied), indicating high satisfaction in these areas. For the remaining questions, an average score of 4.66 (satisfied) was obtained, which also reflects a positive overall evaluation.

Figure 11 shows the boxplot of the survey results from the nurses, indicating that over 80% of the nurses gave an average score between 4.5 and 5 for questions Q1 to Q4 regarding the system's functionality. For Q5 and Q6, 100% of the nurses rated the system's usability category with a 5. Finally, in terms of efficiency, for questions Q7 to Q9, the median responses reached a maximum of 5 and a minimum of 4.68.

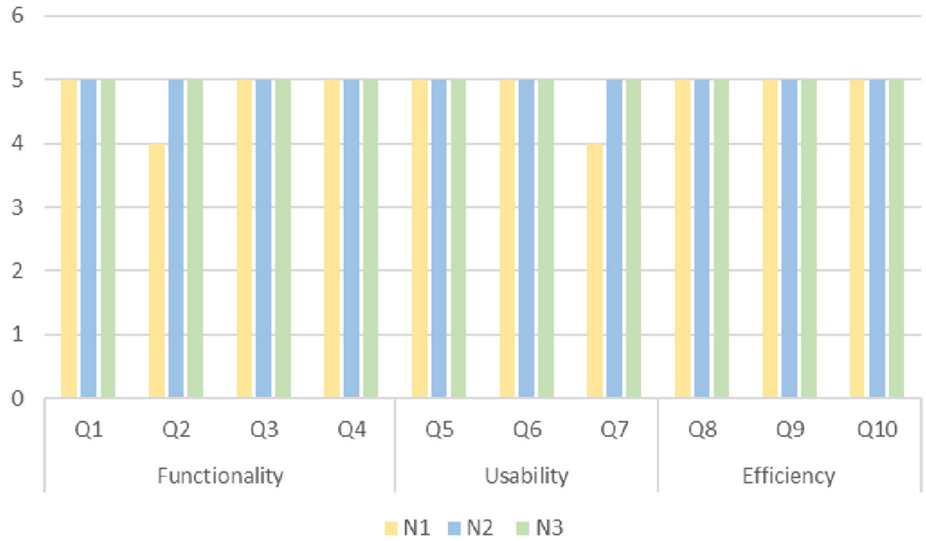

Fig. 10. Survey results for head nurses

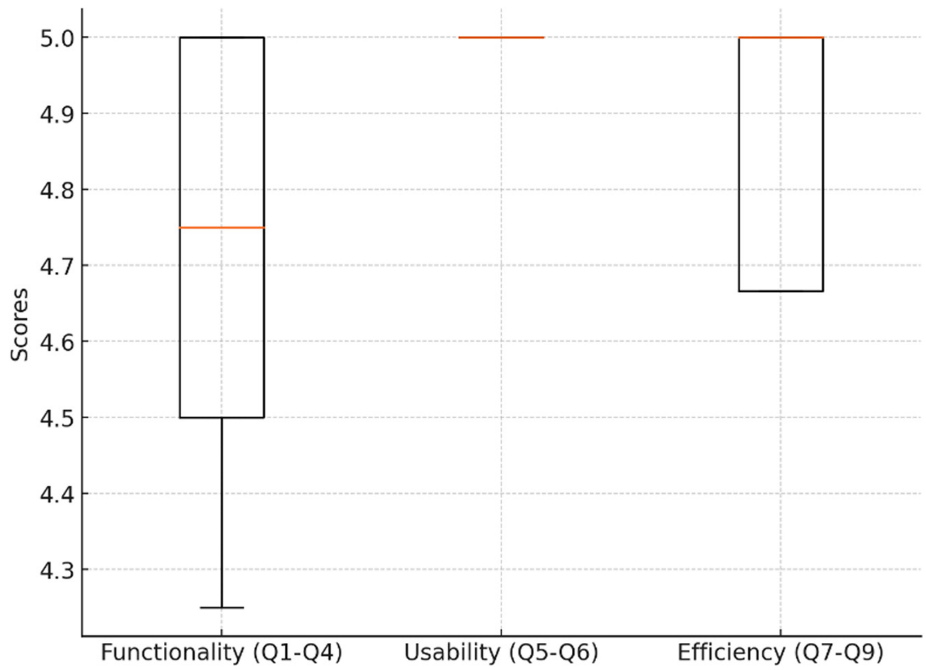

Fig. 11. Boxplot of survey results for nurses

7 CONCLUSIONS

This study proposes an intelligent system based on an RR algorithm and a GA for optimizing nurse scheduling, aimed at reducing the time required for generating schedules. To achieve this, a hybrid algorithm was utilized that integrates the RR algorithm and the GA. Initially, the RR algorithm generates a preliminary schedule that respects nurses' preferences for shifts and days off. This preliminary schedule is subsequently refined by the GA to ensure compliance with hard constraints.

For the preliminary evaluation of the hybrid algorithm, the Virgen del Carmen Maternal and Child Health Centre provided historical records of nursing shifts from previous months. In this analysis, 20 different scenarios were considered, with a total participation of 591 nurses.

The proposal was validated through two experiments conducted with the head nurse and nurses of the Virgen del Carmen Maternal and Child Health Centre. In the first experiment, the weekly schedules of the nurses were generated using the traditional method, involving 50 nurses and three head nurses. In the second experiment, the weekly schedules were generated using the proposed system, with the same participation of 50 nurses and three head nurses.

In the experiment analysis, a significant improvement was observed with Calenurse (Experiment 2) compared to the manual method (Experiment 1). Both achieved 100% shift coverage; however, *Calenurse* showed greater equity in shift distribution with indices of 0.088, 0.126, and 0.113 for Groups 1, 2, and 3, respectively, compared to the higher indices in Experiment 1. Moreover, it met 100% of the hard constraints in all groups, improving the percentages of 90%, 87%, and 85% from Experiment 1. Finally, the schedule generation time was reduced from 40–45 minutes to only 5–9 seconds, demonstrating the high efficiency of *Calenurse*.

For future work, it is recommended that the system address other scenarios, such as testing in clinics and hospitals with a larger or smaller number of nurses, in various departments, and under different constraints. Additionally, it would be beneficial to integrate the system with other hospital modules, such as patient appointment management, to organize schedules according to patient demand, thereby optimizing efficiency and service quality.

8 ACKNOWLEDGMENT

We extend our gratitude to the Research Department of the Universidad Peruana de Ciencias Aplicadas for their support through the UPC-Expost-2024-1 incentive.

9 REFERENCES

[1] R. Otero-Caicedo, C. E. M. Casas, C. B. Jaimes, C. F. G. Garzón, E. A. Y. Vergel, and J. C. Z. Valdés, "A preventive-reactive approach for nurse scheduling considering absenteeism and nurses' preferences," *Operations Research for Health Care*, vol. 38, p. 100389, 2023. https://doi.org/10.1016/j.orhc.2023.100389

[2] S. K. Simić, E. Corchado, D. Simić, J. Đorđević, and S. K. Simić, "A novel fuzzy metaheuristic approach in nurse rerostering problem," *Logic Journal of the IGPL*, vol. 28, no. 4, pp. 583–595, 2020. https://doi.org/10.1093/jigpal/jzz061

[3] A. H. Nobil, S. M. E. Sharifnia, and L. E. Cárdenas-Barrón, "Mixed integer linear programming problem for personnel multi-day shift scheduling: A case study in an Iran hospital," *Alexandria Engineering Journal*, vol. 61, no. 1, pp. 419–426, 2022. https://doi.org/10.1016/j.aej.2021.06.030

[4] M. Hadwan, "Annealing harmony search algorithm to solve the nurse rostering problem," *Computers, Materials & Continua*, vol. 71, no. 3, pp. 5545–5559, 2022. https://doi.org/10.32604/cmc.2022.024512

[5] F. Leung, Y. Lau, M. Law, and S. Djeng, "Artificial intelligence and end user tools to develop a nurse duty roster scheduling system," *International Journal of Nursing Sciences*, vol. 9, no. 3, pp. 373–377, 2022. https://doi.org/10.1016/j.ijnss.2022.06.013

[6] D. Simić *et al.*, "Challenges for nurse rostering problem and opportunities in hospital logistics," *Journal of Medical Informatics & Technologies*, vol. 23, pp. 195–202, 2014. [Online] Available: http://yadda.icm.edu.pl/baztech/element/bwmeta1.element.baztech-91e6de2b-66b9-4de6-884d-5d0ee3c3b7d3

[7] V. R. Muñoz Araya, "Insatisfacción de balance trabajo-vida en médicos y enfermeras de establecimientos de salud de Perú, ENSUSALUD 2016," Universidad Peruana de Ciencias Aplicadas (UPC), 2018. [Online] Available: http://hdl.handle.net/10757/624968

[8] I. Rizany, R. T. S. Hariyati, E. Afifah, and Rusdiyansyah, "The impact of nurse scheduling management on nurses' job satisfaction in army hospital: A cross-sectional research," *SAGE Open*, vol. 9, no. 2, 2019. https://doi.org/10.1177/2158244019856189

[9] C. Sucapuca, W. C. Morales-García, and J. Saintila, "Work-related factors associated with burnout among peruvian nurses," *Journal of Primary Care & Community Health*, vol. 13, 2022. https://doi.org/10.1177/21501319221127085

[10] J. A. Laguna, "Redistribution of salary or professional recognition? The difficult construction of a profession, the Peruvian nursing," *Ciencia & Saude Coletiva*, vol. 25, no. 1, pp. 223–232, 2020. https://doi.org/10.1590/1413-81232020251.25972019

[11] G. Campos Truyenque, V. A. Nolberto Sifuentes, and D. M. Coras Bendezú, "Satisfacción laboral en profesionales de enfermería de un instituto especializado de Perú," *Revista Enfermería Herediana*, vol. 11, pp. 11–17, 2019. https://doi.org/10.20453/renh.v11i1.3520

[12] S. Malekian, A. R. Komijan, A. Shoja, and M. Ehsanifar, "New nurse scheduling model considering nurses' seniority and the possibility of consecutive shifts under COVID-19 (A real case study)," *International Journal of Healthcare Management*, vol. 16, no. 4, pp. 1–19, 2022. https://doi.org/10.1080/20479700.2022.2134639

[13] A. Abayomi-Alli, F. O. Uzedu, S. Misra, O. Abayomi-Alli, and O. T. Arogundade, "Hybrid model of genetic algorithms and Tabu search memory for nurse scheduling systems," *International Journal of Service Science, Management, Engineering, and Technology*, vol. 13, no. 1, pp. 1–20, 2022. https://doi.org/10.4018/IJSSMET.297494

[14] H. Koruca, M. S. Emek, and E. Gulmez, "Development of a new personalized staff-scheduling method with a work-life balance perspective: Case of a hospital," *Annals of Operations Research*, vol. 328, pp. 793–820, 2023. https://doi.org/10.1007/s10479-023-05244-2

[15] A. Turhan and B. Bilgen, "A hybrid fix-and-optimize and simulated annealing approaches for nurse rostering problem," *Computers & Industrial Engineering*, vol. 145, 106531, 2020. https://doi.org/10.1016/j.cie.2020.106531

[16] M. Q. H. Abadi, S. Rahmati, A. Sharifi, and M. Ahmadi, "HSSAGA: Designation and scheduling of nurses for taking care of COVID-19 patients using novel method of hybrid SALP swarm algorithm and genetic algorithm," *Applied Soft Computing*, vol. 108, p. 107449, 2021. https://doi.org/10.1016/j.asoc.2021.107449

[17] Z. Chen, P. De Causmaecker, and Y. Dou, "A combined mixed integer programming and deep neural network-assisted heuristics algorithm for the nurse rostering problem," *SSRN*, 2022. https://doi.org/10.2139/ssrn.4020057

[18] A. Kheiri, A. Gretsista, E. Keedwell, G. Lulli, M. G. Epitropakis, and E. Burke, "A hyper-heuristic approach based upon a hidden Markov model for the multi-stage nurse rostering problem," *Computers & Operations Research*, vol. 130, p. 105221, 2021. https://doi.org/10.1016/j.cor.2021.105221

[19] A. Legrain, J. Omer, and S. Rosat, "An online stochastic algorithm for a dynamic nurse scheduling problem," *European Journal of Operational Research*, vol. 285, no. 1, pp. 196–210, 2020. https://doi.org/10.1016/j.ejor.2018.09.027

[20] 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. https://doi.org/10.1016/j.asoc.2020.106336

[21] X. Zhang *et al.*, "Multi-agent deep Q-network-based metaheuristic algorithm for Nurse Rostering Problem," *Swarm and Evolutionary Computation*, vol. 87, p. 101547, 2024. https://doi.org/10.1016/j.swevo.2024.101547

[22] R. Muniyan *et al.*, "Artificial bee colony algorithm with Nelder–Mead method to solve nurse scheduling problem," *Mathematics*, vol. 10, no. 15, p. 2576, 2022. https://doi.org/10.3390/math10152576

[23] A. Amindoust, M. Asadpour, and S. Shirmohammadi, "A hybrid genetic algorithm for nurse scheduling problem considering the fatigue factor," *Journal of Healthcare Engineering*, vol. 2021, no. 1, p. 5563651, 2021. https://doi.org/10.1155/2021/5563651

[24] S. S. Goh, S. N. Sze, N. R. Sabar, S. Abdullah, and G. Kendall, "A 2-Stage approach for the nurse rostering problem," *IEEE Access*, vol. 10, pp. 69591–69604, 2022. https://doi.org/10.1109/ACCESS.2022.3186097

[25] L. Huang, C. Ye, J. Gao, P. Shih, F. Mngumi, and X. Mei, "Personnel scheduling problem under hierarchical management based on intelligent algorithm," *Complexity*, vol. 2021, no. 1, p. 6637207, 2021. https://doi.org/10.1155/2021/6637207

[26] P. Chen, W. Huang, T. Chiang, and G. K. Chen, "Applying heuristic algorithms to solve inter-hospital hierarchical allocation and scheduling problems of medical staff," *International Journal of Computational Intelligence Systems*, vol. 13, no. 1, pp. 318–381, 2020. https://doi.org/10.2991/ijcis.d.200310.004

[27] A. Turhan and B. Bilgen, "A mat-heuristic based solution approach for an extended nurse rostering problem with skills and units," *Socio-economic Planning Sciences*, vol. 82, p. 101300, 2022. https://doi.org/10.1016/j.seps.2022.101300

10 AUTHORS

Dana Vallejos received degree in Software Engineering from Peruvian University of Applied Sciences in 2024. Her area of development includes front-end development specialized in Angular and mobile development with Flutter (E-mail: u201910362@upc.edu.pe).

Katerin Villalobos received degree in Software Engineering from Peruvian University of Applied Sciences in 2024. Her area of development includes front-end development specialized in React, mobile development with Flutter and algorithms (E-mail: u201913741@upc.edu.pe).

José Luis Castillo-Sequera received degree in computing from the National University of San Marcos, Peru, master's degree in computer project management, master's degree in university teaching in Spain, and Ph.D. in information systems, documentation, and knowledge from the University of Alcalá, Spain. He is currently a Full Professor with the Department of Computer Science, University of Alcalá. He has multiple publications at the level of high-level indexed JCR journals and is the author of books and book chapters related to the field of artificial intelligence (E-mail: jluis.castillo@uah.es).

Lenis Wong is a Professor and researcher of Software Engineering and Information Systems Engineering at the National University of San Marcos and Peruvian University of Applied Sciences, Peru. She holds a PhD in Systems Engineering and Computer Science. MSc. in Systems and Computer Engineering with mention in Software Engineering. She is a member of the AI research group and has carried out different multidisciplinary projects where Artificial Intelligence, Software Engineering and Information Systems Engineering have been applied in education, health, medicine and healthcare. She has published several international peer-reviewed scientific articles in different multidisciplinary areas such as: ML, DL, IoT, e-Health, Software Engineering, Requirements Engineering, Cloud Computing, E-Learning, Gamification, Cyberattacks, Natural Language Processing, Networks and Blockchain Technologies (E-mail: pcsilewo@upc.edu.pe).