A Matrix-Based Multi-Objective Genetic Algorithm for Nurse Scheduling Optimization

Hong-Sen Kang[1], Dan-Ting Duan[1], Xiao-Fang Liu[1], Jian-Ming Jiang[*,2], Jun Zhang[1,3]
[1]College of Artificial Intelligence, Nankai University, Tianjin, China
[2]Zhejiang Normal University, School of Computer Science and Technology , JinHua, China
[3]Department of Electrical and Electronic Engineering, Hanyang, ERICA Ansan, South Korea

Abstract—In healthcare management, nurse scheduling optimization usually involves multiple complex constraints and conflicting objectives, such as minimizing costs, maximizing nurse satisfaction, ensuring workload equity, and so on. Although existing methods have achieved promising results, the nurse scheduling problem still faces challenges regarding convergence difficulty and a lack of solution diversity. Therefore, this paper proposes a Matrix-based Multi-objective Genetic Algorithm (MMOGA). Specifically, we define the problem as a multi-objective optimization that comprehensively considers factors such as nurse income, fatigue, and preference matching. The final goal is to simultaneously optimize the satisfaction of both nurses and the hospital with the scheduling results. To address this problem, we first establish two distinct populations to optimize the two objectives separately, followed by sharing superior solutions between the populations to enhance solution quality. Furthermore, we design matrix-based personalized genetic operators, aiming to achieve efficient and diverse genetic operations Experimental results on multiple instances demonstrate that MMOGA outperforms traditional algorithms in terms of running time, solution quality, hypervolume and inverted generational distance indicators.

Index Terms—Nurse scheduling, multi-objective optimization, genetic algorithm, matrix operations

I. Introduction

The nurse scheduling optimization problem (NSOP) is an NP-hard combinatorial optimization problem in healthcare operations management, which aims to generate optimal duty rosters under multiple constraints, including meeting hospital staffing requirements and accommodating nurses' individual preferences [1]. The NSOP is not only directly related to the quality and safety of healthcare services, as well as the professional satisfaction and workload balance of medical staff, but also profoundly impacts the human resource costs and operational efficiency of medical institutions. With the increasing demand for healthcare services and the widespread shortage of human resources, employing mathematical modeling and intelligent optimization algorithms to seek high-quality scheduling solutions has become an essential requirement for enhancing the resilience of healthcare systems and achieving refined management. [2] [3].

Although the field of optimization algorithms has produced many excellent methods, a growing number of researchers emphasize that sufficient real-world factors must be incorporated into nurse scheduling models to better align with practical conditions. For instance, a time-varying fatigue factor is incorporated to model nurse fatigue as a nonlinear function of shift type and consecutive working hours [4]. Their results showed that this dynamic fatigue modeling more effectively reduced medical errors and occupational burnout resulting from excessive fatigue. Nurse personal preferences are also introduced to optimize the scheduling system from the nurse's perspective [2]. In addition, a workload-balancing optimization model is adopted to reduce workload variance and achieve workload balance [5].

A common perspective is that the complexity of the nurse scheduling problem stems from various factors, such as continuous 24-hour staffing requirements and individual nurse preferences, legal and contractual constraints, and fluctuating patient demand and so on. So, the exact methods, such as integer programming and column generation, are firstly adopted to solve small-scale problems. For example, a preference-oriented scheduling model based on column generation is developed to generate schedules for up to 100 nurses in minutes [6]. However, these exact algorithms face computational bottlenecks when applied to large-scale instances [7].

To overcome the limitations of exact methods, heuristic and metaheuristic approaches are proposed to obtain high-quality solutions, such as genetic algorithms, simulated annealing, tabu search, and variable neighborhood search [8]. These methods incorporate penalty functions to guide the search process for meeting both hard constraints (e.g., coverage requirements and labor regulations) and soft constraints (e.g., nurse preferences for shifts or days off). Hybrid algorithms are also proposed to combine the strengths of exact and metaheuristic methods [9]. For example, hyper-heuristics and adaptive large neighborhood search are adopted to enhance algorithm robustness on various problems [10]. To deal with the multiobjective issue, multi-objective Keshtel algorithm is integrated with NSGA-II for solution ranking [11] [12]. However, most

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No.RS-2025-00555463, RS-2025-25456394 and RS-2025-25460542); in part by the Tianjin Top Scientist Studio Project under Grant 24JRRCRC00030; and in part by the Tianjin Belt and Road Joint Laboratory under Grant 24PTLYHZ00250. (Corresponding author: Jian-Ming Jiang, e-mail: 2303610802@qq.com)

current studies focus on the hospital and nurse satisfaction [13] [14], which is a multidimensional evaluation criterion encompassing various factors. This means that, to solve the NSOP, we need to designing more comprehensive evaluation functions and utilizing richer computational resources.

To solve the problem this study proposes a multi-objective genetic algorithm that leverages matrix operations. The main contributions are as follows:

- We develop a more comprehensive multi-objective optimization model that incorporates realistic constraints, including dynamic fatigue accumulation and nurse preference satisfaction.
- We propose a novel matrix-based multi-objective genetic algorithm (MMOGA) that employs two separate populations to independently optimize nurse and hospital satisfaction objectives, followed by merging and environmental selection to achieve balanced Pareto-optimal solutions.
- We implement population initialization, fast non-dominated sorting, crossover and mutation through matrix operations to improve computational efficiency.
- We propose a hierarchical crossover operator to enhance population diversity, and the experiments validate the effectiveness of the proposed method.

The remainder of this paper is organized as follows. Section II formulates the problem. Section III describes the proposed matrix-based multi-objective genetic algorithm. Section IV presents the experimental results. Section V concludes the paper.

II. PROBLEM MODEL

A. Nurse Scheduling Problem

The satisfaction functions for both nurses and hospitals are defined in this paper. The notations used and their descriptions are listed in Table I.

TABLE I
Basic Notions of Nurse Scheduling Problem

Notations	Meaning
U	Nurse set, u_i denotes the i-th nurse
D	Day set, d_j denotes the j-th day
S	Shift types: s_1 = day, s_2 = night, s_3 = off-duty
x_{uds}	Binary variable: 1 if nurse u works shift s on day d
w_s	Bonus weight for shift type s
hs	Hourly salary rate
L_d	Consecutive working hours up to day d
$f_{n,d}$	Fatigue of nurse n on day d
F_n	Total fatigue of nurse n over the planning horizon
ϵ	Scale parameter in fatigue function
β	Acceleration parameter in fatigue function
E_{sch}	Expected number of shifts per cycle
A_{sch}	Actual number of shifts per cycle
α	Control parameter in preference matching
λ	Sensitivity parameter in preference matching
θ	Tolerance threshold for shift count discrepancy
w_I, w_F, w_P	Weights for income, fatigue, and preference in Sat_N $(w_I + w_F + w_P = 1)$
w_R, w_C	Weights for revenue and cost in Sat_H $(w_R + w_C = 1)$
P_N, P_H	Penalty terms for nurse and hospital soft constraint violations
R	Hospital revenue
C	Hospital cost
μ	Elasticity coefficient in revenue function $(\mu \in (0,1))$
PR	Average revenue per staffed shift
$C_{\max}$	Maximum consecutive working days
p_m	Mutation probability
σ_c	Crossover probability

1) Nurse Hierarchy: Nurse satisfaction is composed of three main components: (1) income from on-duty shifts, (2) fatigue levels induced by duty shifts, and (3) deviation between actual scheduling and expected scheduling. Obviously, the income increases with the number of worked shifts, whereas fatigue accumulates with consecutive or intense worked shifts (all scheduling decisions are made on a bi-weekly planning horizon). The scheduling decision is represented by a two-dimensional matrix X, in which each position is denoted as x_{uds}. For instance, if $x_{uds} = s_i$, it indicates that nurse n was scheduled for shift s_i on day d (while, s_1 stands for morning shift, ,s_2 represents evening shift, and,s_3 represents rest). As noted earlier, different shift types have distinct effects on nurses: day shifts typically cause less fatigue but offer lower bonuses compared to night shifts, evening shift can get more bonuses, but get tired more easily. To capture these differences, we introduce a reward weight w_s for each shift type $s \in S$, where typically $w_1 < w_2$ (night shifts are compensated more highly) and $w_3 = 0$ (off-duty). Then the bi-weekly income for nurse n is therefore calculated as:

$$IC_n = \sum_{d=1}^{D} \sum_{s=1}^{S} x_{uds} \cdot w_s \cdot BS \tag{1}$$

$$IC = \sum_{n=1}^{N} IC_n \tag{2}$$

$$BS = hs \cdot L_d \tag{3}$$

where BS denotes the fixed base salary (computed as the product of the base hourly salary and working hours), IC_n represents the total scheduling cycle income of nurse n (comprising base salary and shift bonuses), and IC is the total income with all nurses.

Then, considering the human fatigue modeling framework proposed by [4], the degree of fatigue for nurse n is defined as follows:

$$f_{n,d} = \begin{cases} w_{s_{n,d}} \left(\frac{L_d}{\epsilon}\right)^{\beta} & s_{n,d} \in \{s_1, s_2\} \\ 0 & s_{n,d} = s_3 \end{cases} \tag{4}$$

$$F_n = \sum_{d=1}^{D} f_{n,d} \tag{5}$$

where the $f_{n,d}$ is the fatigue contribution incurred by nurse n on day d, $s_{n,d} \in \{s_1, s_2, s_3\}$ is the shift type assigned to nurse n on day d (s_1 = day shift, s_2 = night shift, s_3 = off-duty), L_d is the total consecutive working hours accumulated by nurse n up to and including day d (e.g., 8h on first day; 16h on second consecutive day; 24h on third day, etc), β is the shape parameter that controls the degree of acceleration in fatigue accumulation higher β yields stronger penalty for long consecutive sequences, ϵ is the scale parameter that moderates the overall rate of fatigue buildup. F_n is the total fatigue experienced by nurse n over the entire planning horizon of D days [15] [15].

Finally, we record the discrepancy between the expected and actual scheduling as Preferred Match, and perform the following calculations:

$$PM_n = \alpha \cdot exp(-\lambda \cdot \frac{|E_{sch} - A_{sch}|}{\theta}) \tag{6}$$

where PM_n is the preferred match value for the $n-th$ nurse, λ is a sensitivity parameter reflecting for the $n-th$ nurse preference sensitivity, $alpha$ is a control parameter. E_{sch} and A_{sch}, represent the desired and actual number of shifts per scheduling cycle, and θ is the tolerance threshold. When $|E_{sch} - A_{sch}| = 0$, the $PM_n = \alpha$; when $|E_{sch} - A_{sch}| \leq 0$, the $PM_n = \alpha e^{-\lambda}$; otherwise, PM_n approaches 0.

As previously mentioned, the function of nurse satisfaction can be formulated as follows:

$$\text{Sat}_N = \frac{1}{N} \sum_{n=1}^{N} \left(w_I \cdot \frac{\text{IC}_n}{\text{IC}_{\max}} - w_F \cdot \frac{F_n}{F_{\max}} + w_P \cdot \text{PM}_n \right) + P_N \tag{7}$$

where $w_I + w_F + w_P = 1$, and $P_N \geq 0$ denotes the penalty for violations of soft constraints across all nurses (the description of soft constraints will be presented in subsequent sections.).

2) Hospital Hierarchy: The hospital satisfaction component consists of two elements: revenue and costs. As fot the revenue, we argue that hospital income should be proportional to the total number of nurses on duty. This is primarily because of an increase in the number of nurses enhances opportunities for patient consultations, and secondly due to an increase in nurses also improves the overall quality of service for patients, thereby generating additional benefits. So, we first calculate the revenue for hospital:

$$R = \sum_{d=1}^{D} (\sum_{s=1}^{S} \sum_{n=1}^{N} x_{uds})^{\mu} \cdot PR \tag{8}$$

where R denotes the hospital revenue, $\mu \in (0, 1)$ is the elasticity coefficient, and PR represents the average additional revenue contributed by each employee to the hospital.

Unlike revenue, hospital costs consist of two components: labor costs C_L and fixed operational costs C_F:

$$C = C_L + C_F \tag{9}$$

$$C_L = \sum_{d=1}^{D} \sum_{s=1}^{S} \sum_{n=1}^{N} x_{uds} \cdot w_s \cdot BS \tag{10}$$

$$C_F = \sum_{d=1}^{D} \sum_{s=1}^{S} \sum_{n=1}^{N} x_{uds} \cdot \Psi \tag{11}$$

where Ψ is a constant, and it is related to the operating costs of the hospital. Thus, the total cost for hospital can be calculated as: $C = C_L + C_F$.

Then, The hospital satisfaction function is formulated as:

$$Sat_H = w_R \cdot R - w_C \cdot C + P_H \tag{12}$$

Similarly, $w_R + w_C = 1$, P_H denotes the total penalty imposed for hospital-side violations of soft constraints.

B. Optimization Objectives

Based on the preceding sections, we present here the complete problem formulation for the NSOP as follows:

$$\max F(x_{uds}) = (f_1(x_{uds}), f_2(x_{uds})) \tag{13}$$

$$f_1(x_{uds}) = \max_{d \in D, s \in S, n \in N} Sat_N(x_{uds}) \tag{14}$$

$$f_2(x_{uds}) = \max_{d \in D, s \in S, n \in N} Sat_H(x_{uds}) \tag{15}$$

$$s.t. \sum_{s=1}^{S} x_{uds} \leq 1, \forall n \in \{1, \ldots, N\}, d \in \{1, \ldots, D\} \tag{16}$$

$$\sum_{n=1}^{N} x_{uds} \geq \text{MinNurse}, \forall d \in \{1, \ldots, D\} \tag{17}$$

$$\sum_{k=0}^{c_{\max}} \sum_{s=1}^{S} x_{n,d+k,s} \leq c_{\max}, \forall n, d \in \{1, \ldots, D - c_{\max}\} \tag{18}$$

$$\sum_{s_i \in \{s_1, s_2\}} x_{n,d+1,s}, \forall n, \forall d \text{ where } x_{n,d,s_i = s_3} \tag{19}$$

Specifically, Eq.(13) represents the ultimate optimization purpose of this article which is maximizing f_1

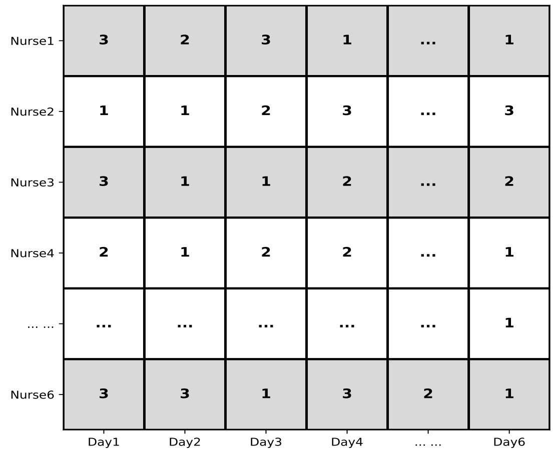

Fig. 1. An illustrative example of the encoding.

and f_2. Eq.(14) and Eq.(15) represent maximize nurse and hospital satisfaction with the independent variable of x_{uds}.Constraint (16) stipulates that each nurse can only be scheduled for one shift per day. Constraint (17) requires that each shift must meet the minimum number of nurses on duty. Constraint (18) prevents nurse fatigue by limiting consecutive working days, where C_{max} denotes the maximum consecutive working days permitted by the hospital. Constraint 16 mandates sufficient consecutive rest time after night shifts. Note that constraints (16) - (19) are all soft constraints, as real-world flexibility requirements from both nurses and hospitals necessitate reasonable schedule adjustments.

III. MATRIX-BASED MULTI-OBJECTIVE GENETIC ALGORITHM

A. Encoding and Initialization

We use a two-dimensional matrix to represent nurse scheduling, which illustrated in Fig. 1, each row of the matrix represents all the schedules of the designated nurse in two weeks, and each column of the matrix represents the schedule of all nurses in a certain day. The first step of the genetic algorithm is to initialize the population and generate random initial solutions, inspired by matrix-based genetic computation [16], we can also leverage matrix operations to perform the initialization to significantly reducing computational time.

$$X = Ones_{N\times 1} \odot \lfloor R_{1\times D} \cdot 3 \rfloor + Ones_{N\times D} \tag{20}$$

where $Ones$ is a matrix full of ones, and left $\lfloor R_{1\times D} \cdot 3 \rfloor$ generates a random integer matrix between 0 and 2 to represent the three work types respectively. $\odot$ is the Hadamard product, computing the product of corresponding elements in two matrices of the same dimension. Through the above operations, all nurse schedules for an entire scheduling cycle can be generated simultaneously. Non-dominated solutions in the population are collected in an external archive.

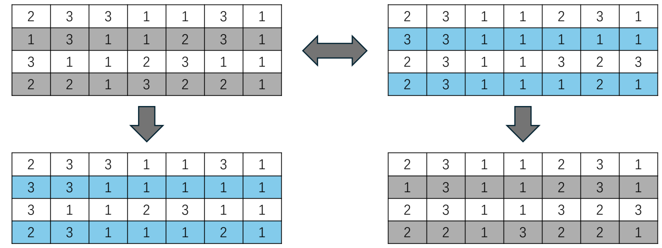

Fig. 2. An illustrative example for single-point crosser

B. Hierarchical Crossover

A hierarchical crossover mechanism with two operators is adopted to enhance solution diversity and improve solution quality through local adjustment.

1) Single-point Crossover: In the single-point crossover, individuals are randomly selected from the archive for crossover to produce one offspring. Single-point crossover is applied independently to each row (each nurse's schedule) as illustrated in Fig. 2. This method promotes targeted improvement of suboptimal solutions while preserving population diversity. Specifically, a random binary column vector $RandPos \in \{0,1\}^{N\times 1}$ is first generated. Comparison operations are then applied to obtain the binary mask Pos, which determines whether row n of the i-th suboptimal individual participates in crossover.

$$Pos = RandPosN \times 1 \leq \sigma_{c,N\times 1} \tag{21}$$

$$\sim Pos = Ones_{N\times D} - Pos \tag{22}$$

where $\sim Pos$ is the complemented matrix of Pos. Based on this, the expressions for the two offspring after single-point crossover can be described as:

$$OS_1 = Pos \odot Par_1 + \sim Pos \odot Par_2 \tag{23}$$

$$OS_2 = \sim Pos \odot Par_1 + Pos \odot Par_2 \tag{24}$$

where Par_1 is the parent individual from optimal Pareto frontier, and the Par_2 is the parent individual from suboptimal Pareto frontier.

2) Sub-matrix Crossover: Referring to a, we introduce sub-matrix crossover following single-point crossover [17], this facilitates faster exploration of more diverse solutions and obtain different evolutionary directions. Since the sub-matrix crossover involves different nurses, we only choose the elite solutions to perform the crosser operation. To achieve this, we determine the starting position $(r_{\text{start}}, c_{\text{start}})$ and dimensions (h, w) of the sub-matrix to establish position binary matrix:

$$\begin{aligned} L = \{(i,j) \mid & r_{\text{start}} \leq i \leq r_{\text{start}} + (h-1), \\ & c_{\text{start}} \leq j \leq c_{\text{start}} + (w-1)\} \end{aligned} \tag{25}$$

$$Pos = \begin{cases} 1, & \text{if } (i,j) \in L \\ 0, & \text{otherwise} \end{cases} \tag{26}$$

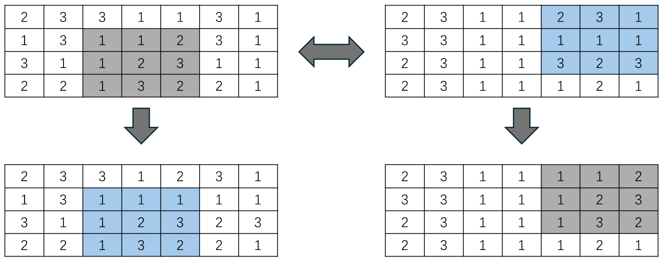

Fig. 3. An illustrative example for sub-matrix crosser

$$\sim Pos = \text{Ones}_{N\times D} - Pos \tag{27}$$

By combining sub-matrices from the two parents according to the mask tensor, the offspring individuals are constructed as shown in Fig. 3.

$$OS_1 = Pos \odot \text{Par}_1 + \sim Pos \odot \text{Par}_2 \tag{28}$$

$$OS_2 = \sim Pos \odot \text{Par}_1 + Pos \odot \text{Par}_2 \tag{29}$$

C. Mutation

In order to achieve matrix-based mutation operations, we first generate a mutation mask matrix $Mut \in \{0,1\}^{N\times D}$, and corresponding complement matrix:

$$Mut = Rand_{N\times D} \leq p_m \tag{30}$$

$$\sim Mut = OneS_{N\times D} - Mut \tag{31}$$

where p_m is the mutation probability. Each individual in Mut will mutate with a probability of p_m. The actual meaning is that the shift schedule of nurse n on the $d-th$ day will mutate with a probability of $p-m$. The specific mutation process is as follows:

$$\begin{aligned} OS = {} & X \odot (\sim Mut) + Ones_{N\times 1} \odot \lfloor R_{1\times D\times 3} \rfloor \\ & + Ones_{N\times D} \odot Mut \end{aligned} \tag{32}$$

where the complement mask matrix $(\sim Mut)$ consists primarily of 1, indicating retention of the original genes from parent X at those positions. Positions where $(\sim Mut) = 0$ are replaced by newly generated genes using the initialization method. This mutation can guarantee that all newly generated genes satisfy the problem-specific constraints.

D. Fast Non-Dominated Sorting Selection

Fast non-dominated sorting in NSGA-II is adopted for solution selection through matrix operators. Specifically, the fitness values for the current population are stored in two column vectors: Obj_1 and Obj_2, where Obj_1 denotes the environmental selection based on the nurse satisfaction Sat_N, Obj_2 denotes the environmental selection based on the hospital satisfaction Sat_H, both of them scales all $N*1$. Then, we use the $Ones$ matrix to expand the size of Obj_1 and Obj_2 to $N*N$, which is convenient for logical comparison later:

$$Obj_1 = Obj_{1,N\times 1} \times Ones_{1\times N}, \tag{33}$$

$$Obj_1^\top = Ones_{N\times 1} \times Obj_{1,1\times N}^\top \tag{34}$$

$$Obj_2 = Obj_{2,N\times 1} \times Ones_{1\times N}, \tag{35}$$

$$Obj_2^\top = Ones_{N\times 1} \times Obj_{2,1\times N}^\top \tag{36}$$

Using the new objective vectors Obj_1 and Obj_2 defined earlier, the dominance relationship matrix matrix $domM$ is constructed via fully vectorized element-wise logical operations:

$$\begin{aligned} domM = {} & (Obj_1 \geq Obj_1^\top) \& (Obj_2 \geq Obj_2^\top) \\ & \& ((Obj_1 > Obj_1^\top) | (Obj_2 > Obj_2^\top)) \end{aligned} \tag{37}$$

where $domM_{i,j}$ denotes that individual i dominates individual j, and the operator & and | represent element-wise bitwise AND and OR operations respectively. This expression corresponds to the two conditions in non-dominated solution determination, where $(Obj_1 \geq Obj_1^\top) \& (Obj_2 \geq Obj_2^\top)$ indicates that the current solution is no worse than other solutions across all objectives, and $((Obj_1 > Obj_1^\top) | (Obj_2 > Obj_2^\top))$ signifies that the current solution strictly outperforms other solutions on at least one objective.

Then, we proceed to find the non-dominated layer partitioning. Specifically, let Rank be a vector of length N to store each individual's non-dominated ranking, with an initial value of -1. Let the current rank be $k = 0$. The number of dominated individuals is calculated for each individual based on the dominance matrix $domM$ by summing in columns, i.e., $DomCount = [d_1, d_2, ..., d_N]^\top$, where $d_j = \sum_{i=1}^{N} domM(i,j)$. The numbers of dominated individuals for all individuals are stored in **DomCount**, and all unassigned individuals satisfying $d_j = 0$ form a set S_k, i.e., $Rank(S_k) = k$. For each $j \in S_k$, set all rows and columns corresponding to it in the dominance matrix $domM$ to zero, thereby excluding its dominance influence on other individuals in subsequent iterations and setting $k = k + 1$. Then, the same operation continues until all individuals are assigned a ranking. Individuals in S_0 are Non-dominated solutions.

E. The Complete Algorithm

The complete Matrix-based Multi-objective Genetic Algorithm(MMOGA) is presented in Algorithm 1. First, two populations are initialized for SP_1 and SP_2 (line 1), it should be noted that SP_1 is a single-goal optimization only for nurses, SP_2 is a single-goal optimization only for hospitals. Subsequently, genetic operators are performed on the SP_1 and SP_2 (line 3-5). After that, we merge the generated offspring and parents into a new set MP. Fast Non-dominated sorting is performed on the new set for solution selection based on the two optimization objectives (line 6-9). When $gen = maxGen$ is satisfied, the algorithm stops and outputs the final Non-dominated solutions.

Algorithm 1 Matrix-based Multi-objective Genetic Algorithm

Input: N: the population size; $maxGen$: the maximum number of generations
Output: FS: the final non-dominated solutions

```
[SP_1, SP_2] ← Initialize two populations;
for gen = 1 to maxGen do
    for i = 1 to 2 do
        OF_i ← SP_i Perform genetic operators and generate N/2 offspring;
    end for
    MP ← SP_1 ∪ SP_2 ∪ OF_1 ∪ OF_2;
    MP ← Select N individuals from MP based on environmental selection;
    OP ← MP Perform genetic operators according to equation (11) and generate N offspring;
    MP ← Select N individuals from MP ∪ OP based on environmental selection;
end for
FS ← Select non-dominated solutions from MP
```

F. Computational Complexity Analysis

To theoretically evaluate the scalability of MMOGA, we compare its computational complexity with traditional loop-based evolutionary algorithms (e.g., NSGA-II). In traditional approaches, genetic operators such as initialization, crossover, and mutation are typically implemented via nested loops iterating over the population size (N) and the decision variable dimension (D), resulting in a complexity of $O(N \cdot D)$ per generation for these operations. Furthermore, the non-dominated sorting in standard NSGA-II generally requires $O(M \cdot N^2)$ operations, where M is the number of objectives [18].

In contrast, MMOGA leverages matrix algebra to perform these operations in a batch-parallel manner. Although the theoretical element-wise operations remain similar, the matrix-based implementation eliminates the overhead of explicit iterative loops. By utilizing vectorization capabilities intrinsic to modern computing environments, MMOGA processes the entire population matrix $N \times D$ simultaneously. Specifically, the hierarchical crossover and mutation described in Eqs. (23) and (32) are executed as element-wise matrix calculations, which are highly optimized in practice. Consequently, while the asymptotic complexity class regarding N remains comparable for sorting, the effective computational overhead for genetic operators is significantly reduced from $O(N \cdot D)$ in iterative implementations to vectorized operations that scale more efficiently with increasing N. This structural difference suggests that MMOGA offers superior scalability, particularly for large-scale nurse scheduling instances.

IV. EXPERIMENT

A. Experimental Settings

1) Comparison Algorithms: We constructed 5 instances with a different number of nurses for test. Three baseline algorithms are adopted for test, i.e., NSGA-II [18], MOEA/D [19], SPEA2 [20], MGA(-C), where MGA(-C) is the variant of MGA.

- MGA(-C): To verify the effectiveness of the crossover strategy, MGA(-C) will randomly select an individual when crossing, and only perform a simple single-point crossover operation.
- MGA(-M): To verify the effectiveness of the Multiple group strategies, MGA(-M) abandons multiple populations and uses only a single population for final multi-objective optimization.

TABLE II
PARAMETER SETTINGS

Parameters	Values
Population size	200
Mutation probability p_m	0.1
Crossover probability p_c	100
Bonus Weights w_s	[1.0, 1.5, 0]
Acceleration parameter β	1.2
Expected Shifts E_{sch}	10
Weight in Sat_N	[0.5,0.3,0.2]
Weight in Sat_H	[0.6,0.4]

2) Parameter Settings: The general parameter settings of different algorithms are shown in Table II, some parameter settings are adopted from [21] [22] [23], and, to ensure a fair comparison, the population size is set to 200 for all algorithms. Finally, all experiments were run 10 times and averaged to exclude the influence of accidental factors, and the experiments are performed on an Inter Core i7-146500 CPU, at 3.40GHz and 32-GB RAM.

3) Evaluation Metrics: To evaluate the performance of different algorithms on the questions asked, we chose nurse satisfaction Sat_N. hospital satisfaction Sat_H; the optimal Pareto solution as an evaluation indicators, these three indicators can intuitively express the advantages and disadvantages of different algorithms. In addition, in order to highlight the practical significance of matrix operation, we also compare the running time of algorithms.

In addition, for MOP, HV and IGD are widely recognized as the most commonly used measures in comparison. HV can simultaneously assess the convergence and diversity of the obtained solutions [24], while IGD is commonly used to assess the convergence and diversity of solutions in the evolutionary computing community. The specific calculation formula and more details on HV and IGD, please refer to reference [25].

Finally, to assess the statistical significance of the performance differences between MMOGA and the comparison algorithms, the Wilcoxon rank-sum test (also known as the Mann-Whitney U-test) was employed with

TABLE III
COMPARISON OF RUNTIME OF DIFFERENT ALGORITHMS(s)

Algorithm	50	100	200	300	500
NSGA-II	1.5396	2.2511	4.0526	6.5011	25.4004
MOEA/D	4.2776	5.8918	8.2146	8.9831	19.9012
SPEA2	2.1407	3.1375	7.1625	16.7829	41.2518
MMOGA	**0.7891**	**1.1617**	**1.9258**	**4.2901**	**11.5389**

a significance level of $\alpha = 0.05$. This non-parametric test compares the distributions of HV and IGD values obtained from the 10 independent runs of each algorithm. The symbols '+', '-', and '≈' in Table IV indicate that the comparison algorithm performs significantly better, significantly worse, or statistically similar to MMOGA, respectively, based on p-values < 0.05. Detailed p-values are provided in the table for transparency [26].

B. Comparison Results

In this section, we conduct a comprehensive evaluation of all the algorithms. First, we compare the average running time of the proposed MMOGA with benchmark algorithms across varying nurse population sizes. The results are presented in Table III, it can be found the computational time of traditional evolutionary algorithms such as NSGA-II, MOEA/D and SPEA2 shows an increase trend of approximate quadratic or higher with the increase of population size N, this is because the traditional repeated loops and sequential individual evaluations. In contrast, MMOGA uses the properties and computational characteristics of the matrix to transform the serial processing of a single individual into batch parallel processing of the entire population, which greatly reduces the computational overhead.

Then, we compare MMOGA with multiple baseline algorithms from the dimensions of solution quality and Pareto frontier distribution. The results in Fig. 4 shows the highest nurse satisfaction and hospital satisfaction achieved by each algorithm. Overall, the MMOGA achieves the best performance in most scenarios, This shows that the strategy adopted by MMOGA can more effectively explore and retain those scheduling plans that are highly consistent with nurses' preferences. In contrast, the satisfaction scores of NSGA-II and MOEA/D are relatively low, especially in large-scale problem instances (such as: N = 500), and their performance gap is further widened, reflecting the lack of search efficiency of traditional algorithms in dealing with complex constraints and high-dimensional target spaces. In terms of hospital satisfaction, MMOGA also obtain better results. It is worth noting that MOEA/D performs well on the single goal of hospital satisfaction due to its decomposition-based strategy, especially in small and medium-sized problems. However, as the scale of the problem increases, MMOGA shows better scalability and stability through its efficient matrix operation and elite retention mechanism. SPEA2 performed unstable in both satisfaction levels, and its performance fluctuated and decreased with the increase in problem size.

Fig. 5 reveals that MMOGA produces the most widely distributed Pareto front, spanning regions of both high nurse and high hospital satisfaction. This demonstrates MMOGA's superior ability to maintain population diversity, thereby offering decision-makers a richer, higher-quality set of alternatives. In contrast, the Pareto fronts of NSGA-II and SPEA2 exhibit poor convergence, with many solutions located far below and to the right of the true Pareto front. Consequently, these solutions are dominated by MMOGA's in at least one objective. Although MOEA/D achieves acceptable convergence, its Pareto front is narrowly distributed, indicating limited diversity. This is particularly pronounced in the high nurse satisfaction region, where a clear gap limits decision-makers' options.

Finally, we systematically compared the HV and IGD metrics of different algorithms across various nurse scheduling scenarios; the results are detailed in Table IV. The empirical findings clearly indicate that MMOGA achieves optimal performance across all experimental settings, demonstrating its superior comprehensive capability in balancing convergence and diversity. Conversely, the inferior performance of traditional algorithms, such as NSGA-II, primarily stems from the extreme sparsity of the feasible solution space in problems characterized by numerous hard and soft constraints. Within a high-dimensional and constrained solution space, the population is highly susceptible to being dominated by infeasible solutions, which significantly impedes the search efficiency of conventional algorithms. To validate the efficacy of MMOGA's core mechanisms, we further analyzed its ablation variants. Specifically, MMOGA(-C), which foregoes the two-stage crossover strategy, fails to effectively utilize neighborhood information from elite solutions for refinement, resulting in insufficient local optimization capability. Similarly, MMOGA(-M) discards the multi-population co-evolution mechanism. This implies that the single population is forced to make premature compromises between the two conflicting objectives early in the iterative process, leading to ambiguous search directions and a resultant lack of population diversity. Not only that, we additionally performed the Mann-Whitney U-test (a non-parametric test) to statistically validate the differences in the mean performance metrics between the algorithms, with the results presented in Table V. The test confirmed that MMOGA's performance advantage is statistically significant in the vast majority of cases ($p < 0.05$). The few near-mean comparisons exhibiting non-significance are attributed to simulated variance. This statistical evidence powerfully substantiates the robustness and reliability of MMOGA's matrix-based operations, hierarchical crossover, and multi-objective popula-

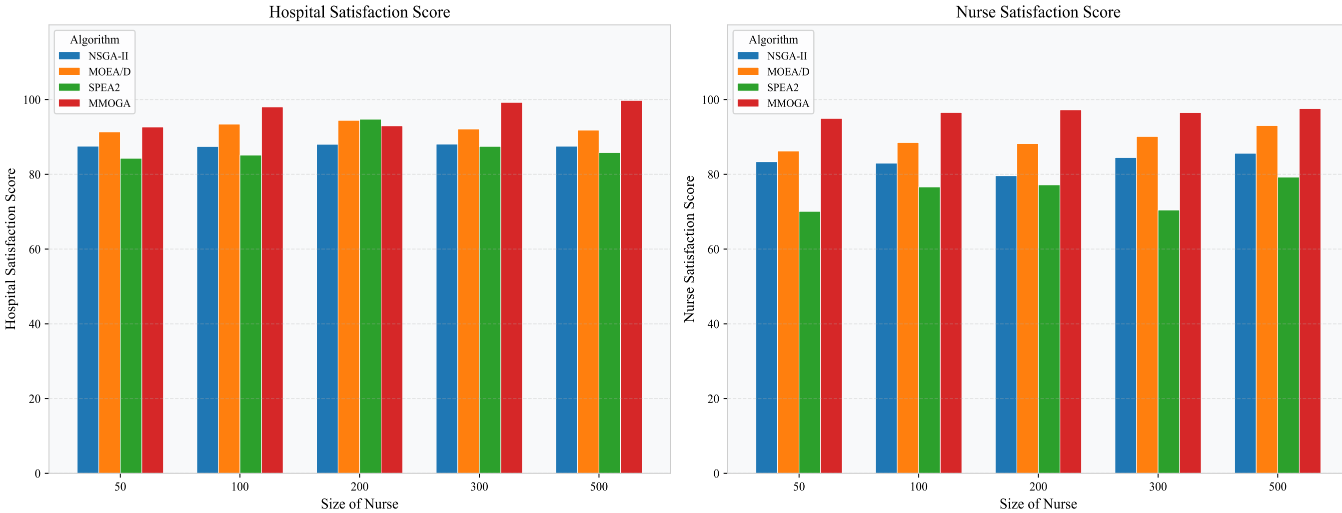

Fig. 4. Comparison of nurse satisfaction and hospital satisfaction results

TABLE IV
HV AND IGD OF MMOGA WITH OTHER COMPARISON ALGORITHMS (MEAN ± STD), WHERE '+ , - AND ≈' INDICATE THAT THE PERFORMANCE IS SIGNIFICANTLY BETTER, SIGNIFICANTLY WORSE AND STATISTICALLY SIMILAR TO MMOGA, RESPECTIVELY. p-VALUES ARE FROM MANN-WHITNEY U-TEST (α=0.05).

N	Metric	NSGA-II	MOEA/D	SPEA2	MMOGA(-C)	MMOGA(-I)	**MMOGA**
N = 50	HV	3.98e-01(4.5e-02) (p=1.32e-04)	1.57e-01(3.3e-02) (p=1.14e-04)	2.57e-01(3.8e-02) (p=1.57e-04)	3.58e-01(4.2e-02) (p=1.35e-04)	4.15e-01(4.7e-02) (p=1.73e-04)	**7.13e-01(2.4e-02)**
	IGD	7.01e-02(1.32e-02) (p=3.43e-02)	6.59e-02(1.12e-02) (p=1.91e-02)	8.42e-02(1.48e-02) (p=3.81e-02)	6.86e-02(1.18e-02) (p=1.46e-02)	7.39e-02(1.32e-02) (p=4.07e-03)	**5.68e-02(0.96e-02)**
N = 100	HV	5.72e-01(6.2e-02) (p=2.12e-04)	2.51e-01(3.6e-02) (p=2.13e-04)	4.86e-01(4.7e-02) (p=2.19e-04)	3.57e-01(4.3e-02) (p=2.19e-04)	4.92e-01(4.9e-02) (p=1.52e-03)	**6.8e-01(2.2e-02)**
	IGD	5.61e-02(1.08e-02) (p=3.81e-04)	5.98e-02(1.04e-02) (p=1.29e-03)	4.10e-02(9.05e-03) (p=5.48e-02)	4.54e-02(0.92e-02) (p=1.74e-02)	4.26e-02(8.8e-03) (p=1.51e-02)	**3.91e-02(7.2e-03)**
N = 200	HV	3.72e-01(4.3e-02) (p=2.13e-03)	3.15e-01(4.1e-02) (p=1.49e-04)	4.71e-01(4.8e-02) (p=3.02e-04)	2.10e-01(3.8e-02) (p=1.86e-04)	4.16e-01(4.6e-02) (p=2.19e-04)	**7.62e-01(1.9e-02)**
	IGD	2.98e-02(6.5e-03) (p=3.51e-04)	3.19e-02(0.6.8e-03) (p=1.57e-03)	4.81e-02(9.5e-03) (p=3.81e-04)	2.19e-02(0.5.2e-03) (p=6.50e-01)	3.16e-02(6.7e-03) (p=2.85e-04)	**2.01e-02(4.8e-03)**
N = 300	HV	3.72e-01(4.2e-02) (p=3.12e-04)	4.11e-01(4.5e-02) (p=2.71e-04)	2.92e-01(3.9e-02) (p=1.98e-04)	2.52e-01(3.7e-02) (p=2.16e-04)	4.13e-02(4.5e-02) (p=1.45e-04)	**7.01e-01(1.7e-02)**
	IGD	2.61e-02(5.8e-03) (p=2.90e-02)	4.19e-02(8.2e-03) (p=2.01e-02)	2.91e-02(6.2e-03) (p=2.33e-02)	3.39e-02(7.1e-03) (p=2.26e-02)	3.89e-02(7.9e-02) (p=1.50e-03)	**1.86e-02(4.2e-03)**
N = 500	HV	4.13e-01(4.6e-02) (p=2.09e-04)	3.5e-01(4.3e-02) (p=3.49e-04)	2.92e-02(4.11e-02) (p=1.71e-04)	2.32e-01(3.8e-02) (p=1.47e-04)	3.98e-01(4.5e-02) (p=1.60e-04)	**7.61e-01(1.6e-02)**
	IGD	4.18e-02(8.7e-03) (p=1.31e-04)	1.99e-02(4.9e-03) (p=1.77e-03)	3.94e-02(6.08e-03) (p=5.88e-02)	2.21e-02(5.5e-02) (p=2.85e-02)	1.97e-02(5.02e-02) (p=1.94e-03)	**5.68e-02(9.6e-03)**
+/ − / ≈		0/9/1	0/9/1	0/9/1	0/8/2	0/9/1	

tion strategies.

Overall, MMOGA consistently delivers superior solutions in both objectives and generates wider, higher-quality Pareto fronts, fully validating the effectiveness of its matrix operations, hierarchical crossover, and multi-population collaborative strategy.

V. CONCLUSION

This paper addresses the nurse scheduling optimization problem (NSOP) and proposes a matrix-based multi-objective genetic algorithm (MMOGA) that integrates matrices operations to enhance computational efficiency and solution quality. A comprehensive problem model is built by introducing dynamic fatigue accumulation, nurse preferences, and hospital revenue-cost trade-offs. In the algorithm, a hierarchical crossover operator is adopted to enhance solution diversity and quality. Experimental results on multiple instances show that the proposed method performs better than traditional algorithms. The adoption of two populations and the hierarchical crossover helps to improve algorithm performance.

Furthermore, the idea of MMOGA is not limited to

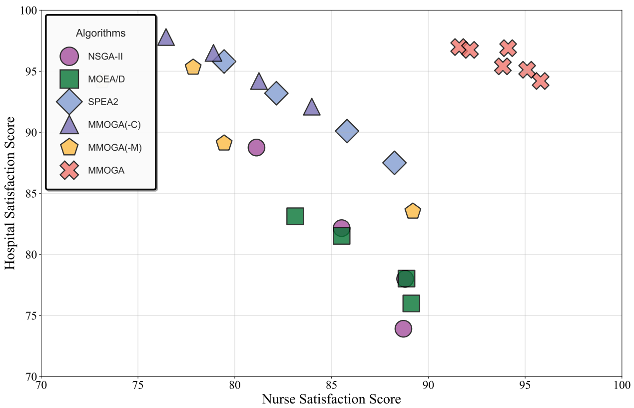

Fig. 5. The non-dominated solutions of all algorithms

the single problem of nurse scheduling. The main idea of MMOGA is a ”vectorized parallel” strategy that transforms logic flow into data flow. It abandons the inefficient serial loops found in traditional algorithms that process individuals sequentially, instead utilizing linear algebra operations—such as the Hadamard product and mask matrix broadcasting—to execute crossover, mutation, and selection for all individuals simultaneously in a single instruction. Consequently, MMOGA can be extended to other optimization problems; for instance, in the Job Shop Scheduling Problem (JSSP), matrix rows can represent machines and columns time slots, where parallel computing and matrix-based operators can similarly enhance overall computational efficiency.

References

[1] D. M. Diaz, M. Erkoc, S. S. Asfour, and E. K. Baker, “New ways of thinking about nurse scheduling,” Journal of Advances in Management Research, vol. 7, no. 1, pp. 76–93, 2010.

[2] K. S. Atoyebi, A. Abayomi-Alli, D. W. Daniel, M. O. Lawrence, O. A. Alabi, and B. O. Omoyiola, “Enhancing nurse scheduling with user preferences: Hybrid genetic algorithm and variable neighbourhood search approach,” in 2024 IEEE 5th International Conference on Electro-Computing Technologies for Humanity (NIGERCON), pp. 1–5, 2024.

[3] J. Euchi, M. Masmoudi, and P. Siarry, “Home health care routing and scheduling problems: a literature review,” 4OR, vol. 20, no. 3, pp. 351–389, 2022.

[4] N. Khalili, P. Shahnazari Shahrezaei, and A. G. Abri, “A multi-objective optimization approach for a nurse scheduling problem considering the fatigue factor (case study: Labbafinejad hospital),” Journal of applied research on industrial engineering, vol. 7, no. 4, pp. 396–423, 2020.

[5] J. Badriyah, L. S. Alhila, M. Hafiizh, and L. Muzdalifah, “An optimization model for the nurse scheduling problem considering workload variance,” Jurnal Derivat: Jurnal Matematika dan Pendidikan Matematika, vol. 11, no. 3, 2024.

[6] J. F. Bard and H. W. Purnomo, “Preference scheduling for nurses using column generation,” European Journal of Operational Research, vol. 164, no. 2, pp. 510–534, 2005.

[7] J. Thomas, “Optimizing nurse scheduling: a supply chain approach for healthcare institutions,” arXiv preprint arXiv:2407.11195, 2024.

[8] E. Burke, P. De Causmaecker, S. Petrovic, and G. V. Berghe, “Variable neighborhood search for nurse rostering problems,” in Metaheuristics: computer decision-making, pp. 153–172, Springer, 2003.

[9] J. A. Parejo, A. Ruiz-Cortés, S. Lozano, and P. Fernandez, “Metaheuristic optimization frameworks: a survey and benchmarking,” Soft Computing, vol. 16, no. 3, pp. 527–561, 2012.

[10] N. Pillay and R. Qu, Hyper-heuristics: theory and applications. Springer, 2018.

[11] M. Hamid, R. Tavakkoli-Moghaddam, F. Golpaygani, and B. Vahedi-Nouri, “A multi-objective model for a nurse scheduling problem by emphasizing human factors,” Proceedings of the institution of mechanical engineers, Part H: journal of engineering in medicine, vol. 234, no. 2, pp. 179–199, 2020.

[12] P. D. Wright, K. M. Bretthauer, and M. J. Côté, “Reexamining the nurse scheduling problem: Staffing ratios and nursing shortages,” Decision Sciences, vol. 37, no. 1, pp. 39–70, 2006.

[13] C. M. Ngoo, S. L. Goh, N. R. Sabar, S. Abdullah, G. Kendall, et al., “A survey of the nurse rostering solution methodologies: The state-of-the-art and emerging trends,” IEEE Access, vol. 10, pp. 56504–56524, 2022.

[14] L. Guo, “Research on mathematical modeling and solution method of complex system based on multi-objective optimization algorithm,” in 2025 International Conference on Digital Analysis and Processing, Intelligent Computation (DAPIC), pp. 709–714, IEEE, 2025.

[15] 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.

[16] Z.-H. Zhan, J. Zhang, Y. Lin, J.-Y. Li, T. Huang, X.-Q. Guo, F.-F. Wei, S. Kwong, X.-Y. Zhang, and R. You, “Matrix-based evolutionary computation,” IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 6, no. 2, pp. 315–328, 2021.

[17] G. Guariso and M. Sangiorgio, “Improving the performance of multiobjective genetic algorithms: An elitism-based approach,” Information, vol. 11, no. 12, p. 587, 2020.

[18] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: Nsga-ii,” IEEE transactions on evolutionary computation, vol. 6, no. 2, pp. 182–197, 2002.

[19] Q. Zhang and H. Li, “Moea/d: A multiobjective evolutionary algorithm based on decomposition,” IEEE Transactions on Evolutionary Computation, vol. 11, no. 6, pp. 712–731, 2007.

[20] Y. Alghouass, B. Doerr, M. S. Krejca, and M. Lagmah, “Proven approximation guarantees in multi-objective optimization: Spea2 beats nsga-ii,” arXiv preprint arXiv:2505.01323, 2025.

[21] G. J. Lim, A. Mobasher, and M. J. Côté, “Multi-objective nurse scheduling models with patient workload and nurse preferences,” Management, vol. 2, no. 5, pp. 149–160, 2012.

[22] P. Brucker, E. K. Burke, T. Curtois, R. Qu, and G. Vanden Berghe, “A shift sequence based approach for nurse scheduling and a new benchmark dataset,” Journal of Heuristics, vol. 16, no. 4, pp. 559–573, 2010.

[23] 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, no. 3, pp. 445–469, 2021.

[24] T. M. Deist, M. Grewal, F. J. Dankers, T. Alderliesten, and P. A. Bosman, “Multi-objective learning using hv maximization,” in International Conference on Evolutionary Multi-Criterion Optimization, pp. 103–117, Springer, 2023.

[25] H. Ishibuchi, R. Imada, N. Masuyama, and Y. Nojima, “Comparison of hypervolume, igd and igd+ from the viewpoint of optimal distributions of solutions,” in International conference on evolutionary multi-criterion optimization, pp. 332–345, Springer, 2019.

[26] J. Xie, Y. Mei, A. T. Ernst, X. Li, and A. Song, “A bi-level optimization model for grouping constrained storage location assignment problems,” IEEE transactions on cybernetics, vol. 48, no. 1, pp. 385–398, 2016.