

A Satisfaction-Based Multi-Objective Mixed-Integer Programming Model for the Nurse Rostering Problem

Noppadon Sakulsom^{1,*}, Pisit Jarumaneeroj²

¹*Department of Mechanical and Industrial Engineering,*

Rajamangala University of Technology Krungthep, Bangkok 10120, Thailand

²*Department of Industrial Engineering, Chulalongkorn University, Bangkok 10330, Thailand*

Received 31 January 2026; Received in revised form 28 February 2026

Accepted 30 March 2026; Available online 30 June 2026

ABSTRACT

A well-balanced working schedule for nurses also means a better patient service quality. Yet, the nurse scheduling problem—also known as a nurse rostering problem—is one of challenging optimization problems, as we need to assign groups of nurses to work functions under a wide variety set of practical constraints. To this end, a nurse may be able to perform limited work functions. Furthermore, the number of assignments must not exceed that allowed by the law—despite the fact that each work function must have at least a predefined number of nurses to perform services. We aim to address this problem, with an overall goal to maximize nurse's quality of life, which is achieved by a multi-objective mixed-integer programming modeling approach. Four different objectives, under different objective orders, are herein considered and the resulting model is solved by means of the epsilon constraint method. With this solution methodology, we are able to determine a group of well-balanced solutions that helps enhance nurse's satisfaction in a long run.

Keywords: Epsilon constraint method; Healthcare; Mixed-integer programming; Nurse rostering; Scheduling; Timetabling problem

1. Introduction

The nurse rostering problem is one of combinatorial optimization problems [1] that has been widely investigated [2]—similar to other problems, such as school

timetabling [3] or production scheduling problems [4, 5]. According to Amindoust et al. [6], a bad nurse roster, with negative aspects—such as understaffing, heavy workload, or insufficient days off—can ad-

versely affect service quality—and so job satisfaction [7, 8]. It may also worsen nurse-patient relationships, or even patient safety.

To address this issue, Miller et al. [2] and Yasmine et al. [8] developed mathematical models to balance between staff coverage and nurse preference. Brucker et al. [9], on the other hand, presented a mathematical model for a similar problem, but with different objective, that is, to minimize penalties of a group of soft constraints to avoid nurse's consecutive working shifts. Finally, Fügener et al. [7] proposed a framework to define nurse cross-training policies to help reduce stress and workload across different units. They also proposed a mixed-integer programming model to examine the possibilities of such policies.

In terms of solution methodologies, a number of algorithms have been devised and applied to the problems. Di Martinelly and Meskens [10], for instance, proposed a heuristic approach that generated a weekly nurse roster, in a team-based fashion, so that affinities between team members were maximized—while nurses' idle time was minimized. Zheng et al. [11] and Lai et al. [12], on the other hand, solved the nurse rostering problems by a randomized variable neighborhood search and a specialized graph construction algorithm, respectively. Besides the abovementioned approaches, a hybrid genetic algorithm was explored by Amindoust et al. [6], while Turhan & Bilgen [1] combined integer programming with Particle Swarm Optimization. Finally, Hu et al. [13] applied a branch-and-price algorithm to solve the nurse rostering problem, given that a nurse could be assigned to multiple units.

Although a number of nurse rostering problems were largely explored in the literature, each of these problems seems to

be unique. This is due to the fact that nurse rosters depend on both restrictions and objectives that vary greatly from one to another unit, hospital, or country. Based on this fact, we herein focus on a nurse roster of a Thai hospital, with an overall goal to maximize the nurse's quality of life, measured by four different metrics.

The remainder of this paper is organized as follows. Section 2 describes the problem. Section 3 presents the methodology that assigns nurses to shifts and functions. Section 4 presents the result and discussions. Finally, Section 5 concludes and suggests future research extensions.

2. Problem Description

In this particular application, the roster of nurses is announced monthly. Yet, before the release of such a roster, each nurse could submit a request for the desired leave days in the roster. Some of these leave-day requests may, unfortunately, be disapproved, as each work function must have enough number of nurses to perform services.

Note that there are two groups of nurses to be assigned in this application, namely registered and assistant nurses. Each group of nurses can work up to a certain number of consecutive working days based on the hospital regulations (i.e., five days for a registered nurse and six days for an assistant nurse).

Regarding nurse assignments, a nurse can be assigned to one out of three shifts in each working day—namely day, swing, and night shifts—each of which covers an eight-hour period. Nonetheless, a nurse could not be assigned with two consecutive shifts, e.g., a night shift followed by a day shift of the next day, to avoid burnout.

In addition to shift requirements,

nurses are also assigned according to work functions based on their skills, such as in-charge nurse, scrub nurse, or anesthesia assistant. Since nurse's skills vary greatly, for ease of scheduling, each nurse is thus assigned with a score ranging from "zero" to "three" for each skill, where "zero" denotes that the nurse could not perform that skill while "three" denotes that the nurse could perform that work function proficiently—or the nurse prefers this work function the most.

In addition to leave days, the number of days off is found important in the literature. According to [14], nurses who had worked two consecutive night shifts were less alert and had less performance. One day off was thus required before bringing them back to a day shift.

We have herein developed a mixed-integer programming model to improve the nurse's quality of life taking into account these constraints. Four different objectives have also been derived to represent different aspects of nurse's satisfaction—and their services—as follows.

- The minimization of violations on leave-day requests.
- The maximization of nurse's work function scores.
- The maximization of consecutive days off.
- The minimization of working hour differences among nurses to improve assignment fairness.

Once set, the resulting multi-objective model is subsequently solved by the Epsilon Constraint Method. In this method, the multi-objective model will be decomposed into four single-objective

models that have been later solved in pre-defined orders. Under a specific objective order, the solutions to the previously solved models will also be placed as constraints in subsequent models with some bounds so that the solutions of subsequent models are less dependent on the previously solved ones [15]. With different objective orders, a group of non-dominated solutions could be generated and used by the schedulers in developing rosters that optimize nurse's satisfaction [12].

The contribution of this study is twofold: the first is to develop a mixed-integer programming model that helps generate nurse roster based on the required conditions and the latter is to explore the differences between solutions from different objective orders.

3. Methodology

A mixed-integer programming model is herein developed to help solve the problem, whose detailed information is provided as follows.

Indices

N = a set of nurses (1, 2, ..., NN). F = a set of work functions (1, 2, ..., NF). D = a set of days (1, 2, ..., ND). D_0 = a set of days, including the last day from the previous month (0, 1, ..., ND). S = a set of shifts (1, 2, ..., NS).

Parameters

H_f = number of working hours of work function $f \in F$. $M \times D_i$ = allowable maximum consecutive working days for nurse $i \in N$. CrD_i = number of consecutive working days of nurse $i \in N$ from the previous month. P_{if} = score of work function $f \in F$ of nurse $i \in N$. $A_{id} = 1$ if nurse $i \in N$ is available on day $d \in D$; 0 otherwise. R_{sfd} = number of nurses required for function $f \in F$ in shift $s \in S$ on day $d \in D$. NR = the number of registered nurses. $M = a$

big integer number.

Decision Variables

$x_{ifsd} = 1$ if nurse $i \in N$ is assigned to perform function $f \in Finshifts \in Sondag$ $d \in D$ $v_{id} = 1$ if the schedule violates the availability of nurse $i \in N$ on day $d \in D$; 0 otherwise. $tO_{id} = 1$ if nurse $i \in N$ on day $d \in D$ has at least two consecutive days off; 0 otherwise. cW_{id} = number of consecutive working days of nurse $i \in N$ on day $d \in D$. cO_{id} = number of consecutive days off of nurse $i \in N$ on day $d \in D$. ctO_{id} = number of consecutive days off of nurse $i \in N$ when there are at least two consecutive days off on day $d \in D$. t_i = total number of working hours of nurse $i \in N$. $m \times R$ = maximum number of working hours of registered nurse. $m \times A$ = maximum number of working hours of assistant nurse.

Objective Function

$$\text{Minimize } z_0 = \sum_{i \in N} \sum_{d \in D} v_{id} \quad (3.1)$$

$$\text{Maximize } z_1 = \sum_{i \in N} \sum_{f \in F} \sum_{(s \in S)} \sum_{d \in D} (x_{ifsd} \cdot P_{if}) \quad (3.2)$$

$$\text{Maximize } z_2 = \sum_{i \in N} \sum_{d \in D} ctO_{id} \quad (3.3)$$

$$\begin{aligned} \text{Minimize } z_3 = & \sum_{i \in 1 \dots NRN} (m \times R \\ & - \sum_{f \in F} \sum_{s \in S} \sum_{d \in D} (x_{ifsd} \cdot H_f) \\ & + \sum_{i \in NRN+1 \dots NN} (m \times A \\ & - \sum_{f \in F} \sum_{s \in S} \sum_{d \in D} (x_{ifsd} \cdot H_f)) \end{aligned} \quad (3.4)$$

Subject to

$$\sum_{f \in F} \sum_{(s \in S)} x_{ifsd} \leq 1 \forall i \in N, d \in D, \quad (3.5)$$

$$\sum_{f \in F} \sum_{(s \in S)} x_{ifsd} \leq A_{id} + v_{id} \forall i \in N, d \in D, \quad (3.6)$$

$$\sum_{(s \in S)} \sum_{d \in D} x_{ifsd} \leq M \cdot P_{if} \forall i \in N, f \in F, \quad (3.7)$$

$$\sum_{i \in N} x_{ifsd} = R_{sfd} \forall f \in F, s \in S, d \in D, \quad (3.8)$$

$$\sum_{f \in F} x_{if1d} = M \cdot (1 - \sum_{f \in F} x_{if3,d-1}) \quad (3.9)$$

$$\begin{aligned} \forall i \in N, d \in 2 \dots ND, \\ m \times R \geq \sum_{f \in F} \sum_{s \in S} \sum_{d \in D} (x_{ifsd} \cdot H_f) \end{aligned} \quad (3.10)$$

$$\begin{aligned} m \times A \geq \sum_{f \in F} \sum_{(s \in S)} \sum_{d \in D} (x_{ifsd} \cdot H_f) \\ \forall i \in NR + 1 \dots NN, \end{aligned} \quad (3.11)$$

$$cW_{i0} = CrD_i \forall i \in N, \quad (3.12)$$

$$\begin{aligned} cW_{id} \leq M \cdot \sum_{f \in F} \sum_{(s \in S)} x_{ifsd} \\ \forall i \in N, d \in D, \end{aligned} \quad (3.13)$$

$$\begin{aligned} cW_{id} + M(1 - \sum_{f \in F} \sum_{s \in S} x_{ifsd}) \geq cW_{i,d-1} + 1 \\ \forall i \in N, d \in D, \end{aligned} \quad (3.14)$$

$$\begin{aligned} cW_{id} \leq cW_{i,d-1} + \sum_{f \in F} \sum_{s \in S} x_{ifsd} \\ \forall i \in N, d \in D, \end{aligned} \quad (3.15)$$

$$cW_{id} \leq M \times D_i \forall i \in N, d \in D, \quad (3.16)$$

$$cO_{i1} = 1 - \sum_{f \in F} \sum_{s \in S} x_{if1s} \forall i \in N, \quad (3.17)$$

$$\begin{aligned} cO_{id} \leq M(1 - \sum_{f \in F} \sum_{s \in S} x_{ifsd}) \\ \forall i \in N, d \in D, \end{aligned} \quad (3.18)$$

$$\begin{aligned} cO_{id} + M \cdot \sum_{f \in F} \sum_{s \in S} x_{ifsd} \geq cO_{i,d-1} + 1 \\ \forall i \in N, d \in 2 \dots ND, \end{aligned} \quad (3.19)$$

$$\begin{aligned} cO_{id} \leq cO_{i,d-1} + (1 - \sum_{f \in F} \sum_{s \in S} x_{ifsd}) \\ \forall i \in N, d \in 2 \dots ND, \end{aligned} \quad (3.20)$$

$$\sum_{f \in F} \sum_{s \in S} (x_{ifs,d-1} + x_{ifs,d}) \leq M(1 - tO_{id})$$

$$\forall i \in N, d \in 2..ND, \quad (3.21)$$

$$ctO_{id} \leq M \cdot tO_{id} \forall i \in N, d \in D, \quad (3.22)$$

$$ctO_{id} \leq cO_{id} \forall i \in N, d \in D, \quad (3.23)$$

$$x_{ifs,d}, v_{id}, tO_{id} \in 0, 1 \forall i \in N,$$

$$f \in F, s \in S, d \in D, \quad (3.24)$$

$$cW_{id}, cO_{id}, ctO_{id}, time_i, m \times R, m \times A \geq 0,$$

$$\forall i \in N, f \in F, s \in S, d \in D. \quad (3.25)$$

Eq. (3.1) is the first objective aiming to minimize the number of times that nurse's leave-day requests have been violated. Eq. (3.2), on the other hand, aims to maximize nurse's work function scores. The third objective, Eq. (3.3), is set so that total number of consecutive days off is maximized, while the fourth objective, namely Eq. (3.4), tries to minimize the deviations on working times among all nurses. Note that, as registered nurses and assistant nurses have different limits of working days, their working hours are, thence, compared among their own groups in Eq. (3.4).

Regarding the constraints, Constraints (3.5) and (3.6) make sure that each nurse is assigned to work only in one shift of a day; and, if a nurse is assigned on a day that (s)he plans to leave, the variable v_{id} will be set to one. Constraint (3.7) states that nurses can be assigned only to work functions that they are capable of, and Constraint (3.8) mandates that the number of nurses assigned to any work function must meet its respective minimum nurse requirement in each shift. Constraint (3.9) prevents nurses from being assigned with two consecutive shifts, i.e., the night shift of a day followed by the day shift of the following day. Constraints (3.10)-(3.11) force variables $m \times R$ and $m \times A$ to be at least the maximum working time of registered nurses and assistant nurses,

respectively.

To account for the workload during the previous month, Constraint (3.12) forces variables cW_{id} at the beginning of the month to be the number of consecutive working days at the end of the previous month, i.e., CrD_i . Constraints (3.13) - (3.15), on the other hand, force cW_{id} to be zero when a nurse does not work on that day and such a variable increases by one when a nurse works on that day. Constraint (3.16) prevents nurses to work consecutively longer than allowable maximum working day.

Constraints (3.17)-(3.20) work in a similar way to Constraints (3.13)-(3.16)—but focus more on consecutive days off. To be precise, these constraints force variable cO_{id} on each day to be zero if a nurse works, or to be one if a nurse does not work. This variable will be increased by one if a nurse has another consecutive day off. Constraints (3.21)-(3.23) force variables ctO_{id} to increase as a nurse has another consecutive day off but the value changes from zero to positive numbers only when a nurse has two consecutive days off. Finally, Constraints (3.24)-(3.25) define the boundary of all decision variables.

With the Epsilon Constraint Method, objective functions are sequentially solved. We need a set of additional constraints for each objective order. Constraints (3.26)-(3.29), for example, form a set of constraints for objective order (1-2-3-4)—i.e., the first objective has the highest priority while the fourth objective is the least prioritized one. In this regard, Constraint (3.26) will force the number of times leave-day-request violations to be lower than a certain limit. Constraints (3.27)-(3.29) work in a similar way, but with some bounds. In case that we use different objective sequences, Constraints (3.26) to (3.29) must be accord-

ingly redefined. Constraints (3.27) to (3.29) force additional objective value to be within epsilon allowable limit. For example, the epsilon (ϵ) is 10% and the objective 1 is to maximize priority, so constraints (3.27) is summation of priority is greater than or equal $(1 - \epsilon)$ or 0.9 multiplied by limit for objective 1. This means the total priority can be as low as 90% of the total priority from previous calculation so the model may find solution with better current objective function. On the other hand, constraints (3.29) allow the minimization objective to be as high as $(1 + \epsilon)$ or 110% of the values from previous calculations.

$$\sum_{i \in N} \sum_{d \in D} v_{id} \leq z_0, \quad (3.26)$$

$$\sum_{i \in N} \sum_{f \in F} \sum_{s \in S} \sum_{d \in D} (x_{ifsd} \cdot P_{if}) \geq 0.9 \cdot z_1, \quad (3.27)$$

$$\sum_{i \in N} \sum_{d \in D} ctO_{id} \geq 0.9 \cdot z_2, \quad (3.28)$$

$$\begin{aligned} & \sum_{i \in \{1..NRN\}} (m \times R - \sum_{f \in F} \sum_{s \in S} \sum_{d \in D} (x_{ifsd} \cdot H_f)) \\ & + \sum_{i \in \{NRN+1..NN\}} (mxA - \\ & \sum_{f \in F} \sum_{s \in S} \sum_{d \in D} (x_{ifsd} \cdot H_f)) \leq 1.1 \cdot z_3. \end{aligned} \quad (3.29)$$

4. Results and Discussions

To assess the proposed model, a group of 10 instances are generated according to the information of a hospital in Thailand. Each instance has 25 nurses, 30 days, 3 shifts, and 6 required functions. Table 1 illustrates an example of input for the underlying nurse rostering problem. Regarding work functions, each function has the required numbers of nurses which may differ across different shifts of weekdays and weekends. Every work function requires an

8-hour period, except for Function 3 which requires a 10-hour period, as shown in Table 1. Also seen from Table 1, each nurse is assigned with a score ranging from “0” to “3” in each function— where “0” means unable to perform and “3” means willing to perform.

The proposed model is solved by CPLEX studio 22.11 with Apple M1 Pro processor and a memory of 16 GB, with a computational time limit of two hours. Note that as there are four different objectives to be solved, but the first objective (i.e., leave-day-request violation) is the most important one, all objective orders will start with this first objective, denoted by OBJ0. And, once solved, the value of this objective will be, later, placed as an additional constraint for subsequent problems.

Table 2 shows a part of solution from the Epsilon Constraint Method, in which the information in each cell indicates the assigned shift (e.g., D, S and N for day, swing and night shifts, respectively) and the assigned work function (e.g., F1 to F6) for each nurse in each day. The cell colors, on the other hand, show whether the assignments violate the leave-day requests. In this regard, the blue color shows the nurse’s assigned days off, while the black one shows the assignments that violate the requested leave days. According to Table 2, Nurse 1, on Day 2, is assigned to perform Function 1 during a day shift. Although Nurse 12 requests a leave day on Day 4, such a request is unfortunately disapproved, as Nurse 12 is assigned to perform Function 4 during the swing shift.

The average computational time for each objective order is shown in Table 3. For ease of comprehension, “Round0” denotes the computation time for solving the problem with the first objective (i.e., leave-day-request violation), while “Round1” to

Table 1. An example of parameters in the underlying nurse rostering problem.

Nurse	Max Day	Current Working Day	Function Priority					
			F1	F2	F3	F4	F5	F6
1	5	2	1	3	1	0	0	0
2	5	2	2	3	2	0	0	0
3	5	0	0	2	1	0	0	0
4	6	2	0	0	0	1	2	1
5	6	0	0	0	0	3	1	1
6	6	2	0	0	0	2	3	1
Number of Required Nurses for Functions	Week day	Day	1	1	1	2	1	1
		Swing	1	1	0	1	1	0
		Night	1	1	0	1	0	0
	Week end	Day	1	2	1	3	1	1
		Swing	1	1	0	2	1	1
		Night	1	1	0	2	0	1
		Working Hour	8	8	10	8	8	8

Table 2. An example of solution from the Ep-silon Constraint Method.

Cell Color		Meaning		Text within cell		Meaning												
		Day off	Working Day	Shift within cell		Shift: D = Day, S = Swing, N = Night												
		Working Day	Working Day (Violated)			Function: Function 1 to Function 6												
Nurse	Day1	Day2	Day3	Day4	Day5	Day6	Day7	Day8	Day9	Day10	Day11	Day12	Day13	Day14	Day15	Day16	Day17	Day18
Nurse1	D,F1			N,F1		D,F1	D,F1	N,F1	S,F1	S,F1	D,F1	N,F1	N,F1	D,F1	D,F1	D,F1	D,F1	D,F1
Nurse2	D,F1	D,F1	D,F1	D,F1	S,F2		S,F1		D,F1	D,F1	D,F1	D,F1		D,F1	S,F1	N,F1		
Nurse3	S,F1	N,F1	N,F2	N,F2		N,F1	S,F1		S,F1	D,F2		N,F1		D,F1	S,F1	D,F1	D,F1	S,F1
Nurse4	S,F1	S,F1	S,F1		D,F1		S,F1		D,F3	S,F2	D,F3	S,F1	S,F1		N,F2	S,F1	D,F1	D,F1
Nurse5	N,F1	N,F1	N,F1	S,F1	S,F1	N,F1		N,F1	N,F1		N,F1	N,F1		N,F1		N,F1	N,F1	N,F1
Nurse6	D,F3	S,F2	D,F3		D,F3	N,F2		N,F2	D,F2	N,F2		D,F2	N,F2	D,F3		N,F2	S,F2	S,F2
Nurse7	D,F3	D,F2	S,F2	D,F2		S,F2	D,F2		S,F2	D,F2		S,F2	D,F2	D,F3	D,F2	D,F2	D,F2	D,F2
Nurse8	S,F2	D,F2		D,F2	N,F2		D,F2	S,F2	D,F2	D,F2		D,F2		D,F2	N,F2	S,F2	D,F2	D,F2
Nurse9	N,F2		D,F2	D,F3	S,F2	D,F2	N,F2		N,F2		N,F2	S,F2		D,F2		D,F2		
Nurse10	D,F2		S,F2		D,F2		D,F3	D,F3	D,F3		D,F3	N,F2	S,F2	N,F2		D,F3		
Nurse11	D,F2	N,F2		D,F2		D,F3		D,F2	S,F2	D,F2		D,F2	S,F2	S,F2		S,F2	D,F2	
Nurse12	S,F4	D,F4		S,F4	D,F4	S,F4		D,F4	D,F4	D,F4	D,F4	D,F4	N,F4			D,F4	N,F4	

“Round3” are those of other objectives in the sequences. For the objective order OBJ213, leave-day-request violation (OBJ0) is first solved in Round0, then total number of consecutive days off (OBJ2) is optimized in Round1, followed by nurse’s work function scores (OBJ1) and deviations on working times among all nurses (OBJ3) in Round2 and Round3, respectively.

It can be observed, from Table 3, that the average time required in later rounds tends to be higher, as more constraints have been placed. Furthermore, total number of consecutive days off (OBJ2) tends to require more time than others as it involves a larger set of constraints with greater number of integer variables.

Average objective values for each ob-

Table 3. Average computational time.

Order	Computational Time (s)				
	Round 0	Round 1	Round 2	Round 3	Total
OBJ 123	0.54	7.54	6502.51	7201.11	13711.70
OBJ 132	0.54	7.54	727.88	7200.96	7936.92
OBJ 213	0.54	5924.20	5854.12	6500.24	18279.10
OBJ 231	0.54	5924.49	7202.12	7201.36	20328.36
OBJ 312	0.54	7.45	7207.13	7201.38	14416.50
OBJ 321	0.54	7.45	7201.12	7201.04	14410.14
Ave	0.54	1979	5782	7084	14847
rage		.73	.54	.31	.12

jective order and their deviations from the best order are shown in Table 4. According to Table 4, the objective values of the least prioritized objectives tend to be worse; and, OBJ3 is typically worse off if it has been solved after OBJ2. This is due to the fact that these two objectives are conflicting. Based on this finding, it could be inferred that the order of objectives plays a significant role in creating good rosters. And, since the objective order could impact on the satisfaction of nurses, decision makers may need to consider the objective orders that give overall lower differences such

as OBJ132 or OBJ312.

Table 4. Average objective values and difference from the best-found values.

Seq uence	Average Objective Value			Difference from Best (%)		
	OBJ1 (Max Satis faction)	OBJ2 (Max Day Off)	OBJ3 (Min Work Diff)	OBJ1 (Max Satis faction (%))	OBJ2 (Max Day Off (%))	OBJ3 (Min Work Diff (%))
OBJ 123	1,112 .70	995	524	2	2	589
OBJ 132	1,102 .60	612	78 .20	3	40	3
OBJ 213	1,135 .50	1,013 .70	1,186	0	0	1461
OBJ 231	1,081 .30	729 .70	142 .20	5	28	87
OBJ 312	1,086 .80	626 .10	76	4	38	0
OBJ 321	1,083 .40	620 .20	76	5	39	0
Ave rage	1,100 .38	766 .12	347 .07	3	24	357
Best	1,135 .50	1,013 .70	76	0	0	0
Worst	1,081 .30	612	1,186	5	40	1461

To further visualize these four different aspects from Round1 to Round3, we have normalized the objective values found in each computational round—so that they range between “0” and “100”, where “0” and “100” are the worst and the best found objective values—and, later, placed such a solution in a radar charts, as shown in Fig. 1.

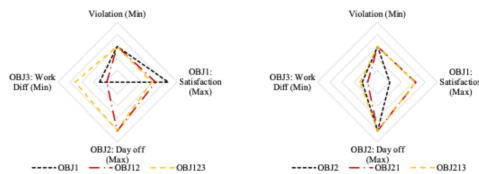


Fig. 1. Objective values from the first to last steps of sequences OBJ123 and OBJ213.

From Fig. 1, the solution to objective order OBJ123 is shown on the left and that of objective order OBJ213 is shown on the right. It can be seen from Fig. 1 that, from the first to the last computational round (e.g., the black, red and yellow lines), the value of OBJ0 remains the same, but

other objective values change to a more balanced shape.

Another aspect to consider is how availability of nurses affects the solution (i.e., roster). We investigate this by dividing the instance into three groups, according to different levels of nurse availability (i.e., 80%, 85% and 90%). In this regard, if a nurse is available for 24 days out of 30 days, such an instance will be recognized as an 80-percent availability group. Under objective orders OBJ123 and OBJ321, the normalized objective values, under different nurse availability, are shown in Fig.2. From Fig. 2, it is obvious that higher availability leads to better OBJ0 values.

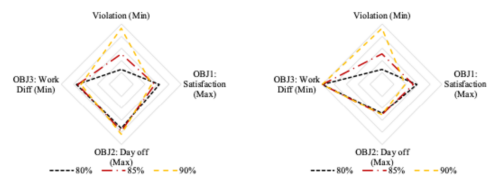


Fig. 2. Availability of nurses and the objective values.

5. Conclusion

The nurse rostering problem has been studied for decades, but with different restrictions and objectives depending on the focus of researchers. In this research, we focus on nurse’s quality of life, in which a mixed-integer programming model is developed and solved by means of the Epsilon Constraint Method.

According to four different satisfaction aspects—namely the violation on nurse’s leave-day requests, the satisfaction of nurses on the assigned functions, the number of consecutive days off, and the deviations of working hours among nurses—the Epsilon Constraint Method could generate solutions with trade-offs among these different aspects. We find that the aver-

age times required in later rounds tend to be higher and maximizing the total number of consecutive days off (OBJ2) tends to require more time, compared to other objectives. We also find that higher availability generally leads to rosters with better total number of leave-day-request violation values (OBJ0).

Hospital managers or schedulers might treat OBJ1 to OBJ3 on the same level of importance. In that case, we would suggest solving the problem in sequences of OBJ312 or OBJ321 as they can give well-balanced solutions with quite low difference of objective values from the best values.

As a concluding remark, there are some limitations worth mentioning. Firstly, the problem is long known to be an NP-hard one, bigger problems may not be practically solved in a reasonable time period. Hence, an efficient heuristic is therefore in need so that we can determine a good pool of non-dominated solutions. Also, rescheduling is likely in a real-life application. As such, the devised heuristic should be able to support re-rostering with the least impact on the current roster.

References

- [1] Turhan AM, Bilgen B. A mat-heuristic based solution approach for an extended nurse rostering problem with skills and units. *Socio-Econ Plan Sci.* 2022;82:101300.
- [2] Miller HE, Pierskalla WP, Rath GJ. Nurse scheduling using mathematical programming. *Oper Res.* 1976;24(5):857–870.
- [3] Sakulsom N. Minimizing distance between classes in school timetabling problem using a mixed-integer programming model. *Adv Transdiscipl Eng.* 2023;35:767–775.
- [4] Pandey P, Gajjar H, Shah BJ. Staffing, tour scheduling, and re-planning in specialty stores with regular and on-call workers. *Comput Ind Eng.* 2025;204:111089.
- [5] Frihat M, Hadj-Alouane AB. A branch-and-cut algorithm for the integrated employee timetabling and hybrid job-shop scheduling problem with time lags and setup times. *Comput Ind Eng.* 2025;208:111351.
- [6] Amindoust A, Asadpour M, Shirmohammadi S. A hybrid genetic algorithm for nurse scheduling problem considering the fatigue factor. *J Healthc Eng.* 2021;2021:5563651.
- [7] Fügener A, Pahr A, Brunner J. Mid-term nurse rostering considering cross-training effects. *Int J Prod Econ.* 2018;196:176–187.
- [8] Yasmine A, Yassine O, Farouk Y, Hicham C. Workload balancing for the nurse scheduling problem: A real-world case study from a French hospital. *Socio-Econ Plan Sci.* 2024;95:102046.
- [9] Brucker P, Qu R, Burke E. Personnel scheduling: Models and complexity. *Eur J Oper Res.* 2011;210:467–473.
- [10] Di Martinelly C, Meskens N. A bi-objective integrated approach to building surgical teams and nurse schedule rosters to maximise surgical team affinities and minimise nurses' idle time. *Int J Prod Econ.* 2017;191:323–334.
- [11] Zheng Z, Liu X, Gong X. A simple randomized variable neighbourhood search for nurse rostering. *Comput Ind Eng.* 2017;110:165–174.
- [12] Lai DSW, Leung JMY, Dullaert W, Marques I. A graph-based formulation for the shift rostering problem. *Eur J Oper Res.* 2020;284:285–300.

- [13] Hu W, He X, Luo L, Pardalos PM. A branch-and-price approach for the nurse rostering problem with multiple units. *Comput Ind Eng.* 2024;198:110629.
- [14] Chang YS, Wu YH, Chen HL, Hsu CY. Is one day off sufficient for re-adaptation to a daytime routine after two consecutive nights of work? *Ergonomics.* 2018;61(1):162–168.
- [15] Fathollahi-Fard AM, Wu P, Tian G, Yu D, Zhang T, Yang J, Wong KY. An efficient multi-objective adaptive large neighborhood search algorithm for solving a dis-assembly line balancing model considering idle rate, smoothness, labor cost, and energy consumption. *Expert Syst Appl.* 2024;250:123908.