Simulation for Dynamic Patients Scheduling Based on Many Objective Optimization and Coordinator

Ali Nader Mahmed, M. N. M. Kahar*
First Faculty of Computing, College of Computing and Applied Sciences Universiti Malaysia Pahang, Malaysia
E-mail: mnizam@ump.edu.my
*Corresponding author

Keywords: dynamic, hospital admission and scheduling, patients' admission scheduling, multi-objective optimization, many objective optimizations, non-dominated optimization

Received: October 4, 2023

The Patient Admission Scheduling Problem (PASP) involves scheduling patient admissions, hospital time locations, to achieve certain quality of service and cost objectives, making it a multi-objective combinatorial optimization problem and NP-hard in nature. In addition, PASP is used in dynamic scenarios where patients are expected to arrive at the hospital sequentially, which requires dynamic optimization handling. Taking both aspects, optimization and dynamic utilization, we propose a simulation for dynamic patient scheduling based on multi-objective optimization, window, and coordinator. The role of multi-objective optimization deals with many soft constraints and providing a set of non-dominated solution coordinators. The role of the counter is to collect newly arrived patients and previously unconfirmed patients with the aim of passing them on to the coordinator. Finally, the role of the coordinator is to select a subset of patients from the window and pass them to the optimization algorithm. On the other hand, the coordinator is also responsible for those selected from the non-dominant solutions to activate it in the hospital and decide on unconfirmed employees to place them in the window for the next round. Simulator evaluation and comparison between several optimization algorithms show the superiority of NSGA-III in terms of set criticality and soft constraint values. Therefore, it treats PASP as a multi-objective dynamic optimization of a useful solution. NSGA-II is guaranteed 0.96 percent dominance over NSGA-II and 100 percent dominance of all other algorithms.

Povzetek: Gre za dinamično razporejanje pacientov z uporabo večciljne optimizacije, ki obravnava kompleksni problem razporejanja sprejema pacientov v bolnišnico, izboljšuje kakovost storitev in učinkovitost z uporabo NSGA-II algoritma za optimizacijo.

1 Introduction

In the 21st century, life expectancy doubled globally, and new health delivery models and technologies are predicted to considerably extend healthy life expectancy [1]. The demand for healthcare services has risen in recent decades because of an ageing population and advancements in preventative care [2], yet the healthcare sector is still under pressure. to reduce costs and raise standards of treatment. The healthcare industry has mainly shifted its focus to a value-based strategy to offset a potential increase in clinical medicine expenditures that are not accompanied by appreciable improvements in health outcomes [3]. In this situation, achieving the greatest results at the lowest cost is the main objective, making effective resource management and patient happiness crucial but competing goals that health care administrators must meet. Practical concerns including admissions control, process design, aggregate planning, capacity distribution, and appointment scheduling must be taken care of in order to address this obstacle. The Patient Admission Scheduling Problem is one of these issues (PASP). Patients' admission scheduling problem (PASP) is how to plan patient's admission and their location and time in the hospital in order to meet certain quality of service and cost objectives [4]. It is considered as complex combinatorial optimization process with many constraints [5]. This is because it involves allocating resources for patients according to the condition of the hospital and the condition of the patient in order to meet the satisfactory level of the patient within the time limit for scheduling. Choosing an appropriate room to allocate to patients while taking into account medical needs, patient demands, and hospital resource availability is the focus of the patient bed assignment problem (PBAP), a PASP sub-task [6]. It is considered as a paramount problem for hospitals and medical centers. PBAP is an NP-hard problem [7]. For solving PBAP, it is needed to create an autonomous system that receives patients requests online or through phone and automatically assign them to beds without the need of human intervention. A conceptual representation of this process is depicted in Figure 1 and the result is mapping patients to the best beds inside the rooms for meeting both the health and satisfaction requirements.

Patient Scheduling is regarded as constrained combinatorial optimization problem with NP-hard nature. Adding the dynamic in terms of patient's arrival and change of preference to the problem makes more complex.

In addition, the problem has a limitation in terms of capacity of the room which leads to a condition of over-crowding that needs to be minimized. Another added complexity to the problem is the need to identify various information of the patients' conditions, their special need and the criticality of their cases before performing the mapping. The process should be automated in order to facilitate the management of the hospitals and to increase the quality of service within the allocated cost.

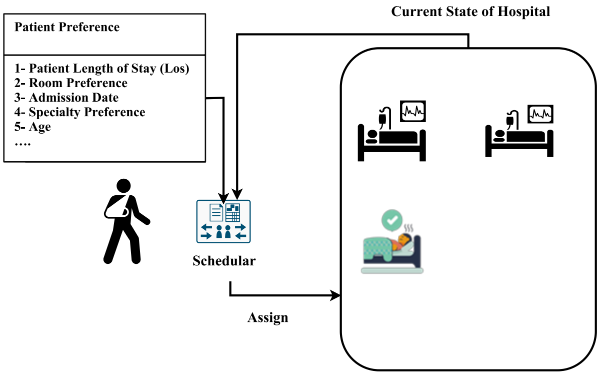


Figure 1: Conceptual representation of the process of automatic PBAP in hospitals.

Meta-heuristic searching optimization algorithms are set of optimization algorithms with capability of solving complex optimization problem based on generating candidate solutions randomly and enabling an evolving of them based on heuristics [8]. The literature contains wide range of meta-heuristic optimization algorithms, some of them are inspired from biological phenomena such as genetic algorithm [9], others are inspired from physical phenomena such as simulation annealing [10]. In addition, there is numerous metaphors used for deriving meta-heuristic algorithm such as ant colony [11], artificial bee colony [12], particle swarm optimization [13]…etc. Despite the type of the metaphor, we can classify the meta-heuristic optimization algorithms into two main categories: single objective and multi-objective [14]. In the single objective, the algorithm aims at optimizing a formulated a single objective function from the problem definition while in the multi-objective, the algorithm aims at optimizing simultaneously multi-objective functions using the concept of Pareto domination. The latter type can be utilised to solve PBAP by treating soft-constraint violations as multi-objective functions that must be minimised during optimization [15]. A strategy for enabling the algorithm to take into account the dynamical nature of the problem must be developed before a multi-objective meta-heuristic optimization algorithm may be used directly. In this article, we propose a simulation that extends the optimization with additional steps in order to enable dynamic scheduling for PBAP using multi-objective optimization. In addition, we provide an algorithm for selecting one solution of the pareto front to use it for providing the allocation decision under two sets: confirmed allocated patients and non-confirmed allocated patients.

The rest of the article is divided into the following sections. We present the contribution in section 2. Next, the literature survey is presented in section 3. Afterwards, a background is provided in section 4. In addition, we present the methodology in section 5. Next, experimental works and evaluation are provided. Lastly, the conclusion and future work are provided in section 7.

2 Contribution

The development of dynamic patient scheduling that supports many patient objectives is the ultimate purpose of this study. The contributions listed below are presented in this article.

1. To the best of our knowledge, this article provides the first in terms of simulating arrival of patients to hospital and an algorithm for scheduling using multi-objective optimization, solutions selection, and scheduler.
2. The scheduling in this article avoids implicit constraint that causes greedy behavior by using the concept of non-confirmed patients. More specifically, it provides list of non-confirmed patients which automatically feeds another list of confirmed patients when their scheduling day is within less than D days on one side and provides the remaining patients inside list of non-confirmed patients to a new call of optimization on the other side.
3. This article enables dynamic multi-objective optimization through solution selection. More specifically, considering that multi-objective optimization algorithm provides Pareto front which represents set of non-dominated solutions, one solution is to be selected for enabling or scheduling. In order to do so, the algorithm performs solution selection using weighted summation of the objectives with respect to their corresponding preference.
4. In order to distinguish between patients that are allowed for rescheduling from new arrived patients, we use variable length optimization (VLO). In VLO, different lengths of solutions are used where each solution allow for rescheduling of different sub-sets of the non-confirmed patients.

3 Literature survey

There are two subsections in the literature. The first is the patient admission scheduling literature, which is discussed in sub-section 2.1. The second is discussed in sub-section 2.2 and is about the application of multi-objective optimization methods to scheduling problems.

3.1 Patient admission scheduling

In the work of [16] which has aimed at solving the problem of PAS based on offline perspective. His proposed combinatorial formulation of the optimization problem of PAS using integer linear programming and proposed Tabu search algorithm for solving it. They aimed at finding the optimal bed assignments for elective patients based on pre-knowledge of the hospital departments, rooms capacity, beds availability, equipment, technical issues, and qualitative elements like the patient's choice for gender, age, and room compatibility. Their work has drawn criticism from a number of angles, including the impracticality of an offline solution given the dynamic nature of the issue, and considering optimizing a weighted average of the soft-constraints which can cause sub-optimality due to the non-convexity of the model or limit the choices to the decision maker due to providing only single optimal solution. In the work of [17], Fix-and-Relax (F&R) and Fix-and-Optimize (F&O) are techniques based on Mixed Integer Programming (MIP) that break down PAS problem instances into smaller chunks before optimising the smaller chunks. More specifically, iteratively improved Quick solutions produced by the F&R heuristic are fed into the F&O heuristic. Patient length of stay (LoS), room preference, admission date, specialty preference, age, as well as time decomposition taking different optimization window sizes, are the factors that have employed decompositions. Ceschia and Schaerf (2016) suggested a different formulation for the demester problem known as a dynamic patient-to-room assignment problem that helped reduce the number of decision variables, compute different lower bound values by omitting some constraints, and adapt simulated annealing to find the best solution. The work of [16] has also been improved by [18] by including local search moves into two tiers of heuristics or hyper-heuristics. The great deluge algorithm was used in this work as a component of the hyper-heuristic, but it was criticised in the work of [19] due to the linear decay rate of its deluge algorithm, which was improved to non-linear adaptive decay rate using the same soft and hard constraints of demester [16] . The scheduling goals in the work [20] were divided into short- and long-term goals, and periodic re-optimization was employed. Using column generation and Dantzig-Wolfe decomposition, the lower bounds are computed. A scheduling algorithm is used in the research of [21] to schedule tourist travel to destination medical centres. The goals are to keep patients' preferred commencement days and flow times as close to real time as possible. They scheduled everything using a flow-shop system. Simulated annealing and tabu search were used with simulation for optimization. The simulation is based on discrete event simulation, which assesses the solution considering the admission day, admission time, and patient sequence as decision factors on each day. The current deterministic model created by [16] was modified in the work of [22] to become stochastic. To represent the arrivals and departures, they employed discrete phase type distribution and a Poisson distribution, respectively. Hence, their model has evolved from the previous deterministic one into a stochastic one. The work of [9] involved the modelling of appointment times that depend on both the needs of the patients and the speed factor of the doctors' performance. Their model is solved utilising a genetic algorithm for large-scale problems and a single solver for small-scale problems. Overall, the literature has addressed the PAS issue from a variety of angles and levels of practicality, including the addition of soft limits, the unpredictability of LoS, and the acceptance of urgent patients. However, the non-domination component of the issue has not been addressed by any of the prior solutions. When dealing with the soft restrictions as separate objectives, the PAS problem is a multi-objective optimization problem. In this manner, By using the penalty concept, we give the decision-maker more options and reduce the disadvantage of the linear combination of soft constraints under the weighted average formula, taking into account that the latter has no application in the problem and the linear combination of constraints does not correspond to the real-world model. Table 1 presents an overview of the existing approaches of PAS in the literature with the key features and criticism and improvements. Table 2 provides a summary of the various methods for patient bed scheduling.

Table 1: Summary of patient admission scheduling (PAS) approaches in literature

Reference	**Approach/Technique**	**Key Features**	**Criticisms/Improvements**
[9]	Genetic Algorithm	Appointment times based on patient needs and doctor performance	Provides single solution
[16]	Integer Linear Programming & Tabu Search	Offline solution, optimal bed assignments considering various constraints	Criticized for impracticality in dynamic settings; limited by single optimal solution
[17]	Fix-and-Relax (F&R), Fix-and-Optimize (F&O)	Decomposition of PAS problem, Mixed Integer Programming	Subject to local minima because of decomposition
[18]	Hyper-heuristics & Great Deluge Algorithm	Improved [16] by local search moves	Criticized for linear decay rate in deluge algorithm
[19]	Non-linear Adaptive Decay Rate	Improved [18] using non-linear adaptive decay rate	Not handling dynamic environment
[20]	Column Generation & Dantzig-Wolfe Decomposition	Short- and long-term scheduling goals, periodic re-optimization	Not handling dynamic environment
[21]	Flow-shop System, Simulated Annealing & Tabu Search	Scheduling for tourist travel to medical centres	Local search capability only
[22]	Stochastic Model	Discrete phase type distribution, Poisson distribution	Evolution from deterministic to stochastic model
[23]	Dynamic patient-to-room assignment	Reduced decision variables, simulated annealing	It does not have global search capability

Table 1: Overview of the various approaches for patient, bed scheduling.

Author	Hard constraints	Soft-constraint	Objective function	Algorithm	Limitation
[16]	8	5	Weighted average	Tabu Search	Sub-optimality due to weighted average and non-convexity
[24]	2	4	Weighted average	Simulated annealing	Weighted average causes sub-optimal result
[25]	3	6	Weighted sum	Hyper-heuristic	Weighted average causes sub-optimal result
[26]	5	3	Weighted sum	deluge algorithm	Weighted average causes sub-optimal result
[27]	-	8	Weighted sum	Mixed Integer Programming (MIP)	more computational time
[28]	12	2	Weighted sum	tabu search (TS) and simulated annealing (SA) with simulation	Not including resource utilization, age and gender
[9]	15	4	Weighted sum	Genetic algorithm	Concern about convergence, sub-optimality due to weighted sum

3.2 Multi-objective optimization for scheduling

Various scheduling issues and applications have been solved using the multi-objective particle swarm optimization technique. Modified multiple-objective particle swarm optimization (MMOPSO), which was proposed by Ghasemi, Khalili-Damghani, et al. in 2019, was used to solve a mixed-integer mathematical programming model for the earthquake reaction phase. Two local search operations are included in the improved multi objective particle swarm optimization. The model considers two target functions: lowering the total cost of facility location and allocation, as well as decreasing the amount of supply deficit. This method beat out the two well-known non-dominated sorting genetic algorithms, NSGA-II and epsilon constraint method, in tests. In the study of Adhikari and Srirama (2019), a modified variation of multi-objective particle swarm optimization was used to optimise the problem of container-based scheduling for the Internet of Things in a cloud context. Energy usage and computing time are the two optimization goals that the writers have considered. To assess the quality of the solution, the weighted sum approach-based fitness function is used to cope with the multi-objective elements.

The acceleration component of multi-objective particle swarm optimization changed the convergence speed. Considering that the typical PSO looks for the best possible solution by combining the individual and current global bests of the particles the acceleration PSO (APSO) approach, which is a modification of the PSO algorithm based on its velocity and displacement, was developed in (Yang, Deb et al. 2011) due to the limits of convergence speed and accuracy. The APSO approach lowers unpredictability as iterations continue by using the individuals that perform best globally. In the study by Fang and Popole (2019), which generated neighborhoods for each particle and used the self-organizing mapping (SOM) approach to select the neighborhood best solution, the particle swarm optimization was modified once again to enhance its searching performance. Analytical research of the convergence of self-adaptive PSO (APSO) with the purpose of presenting a parameter selection method that ensures the convergence was carried out in the work of [29]. Using the suggested SAPSO, they created the SAMOPSO MOO framework, which is based on SAPSO. They also create an external repository that stores the non-dominated solutions in order to obtain a well-distributed Pareto front. The proposed MOO system then uses a cyclic sorting mechanism to update the external repository while integrating elitist-preserving principles. Particle swarm optimization has been modified in the work of [30] to tackle large dimensional discrete variables. To enhance the performance, the method included stretching and changing neighborhood search techniques. Jumping PSO, variable neighborhood search, and the stretching approach are all included in their whole integrated model. Non-dominated sorting genetic algorithm was slightly adjusted and used to solve the scheduling of surgeries in operating rooms in the work of [31]. This work shows that the modification of the searching algorithm is not limited to particle swarm optimization method. The resolved model is a resource allocation methodology that primarily concentrates on allocating operating rooms (ORs) for each surgical specialty (SS). The initialization of the population and the selection using the tournament comprised the first part of the change to NSGA-II. An idea for a multi-patent crossover genetic algorithm appeared in the publication [32]. When it functions for n parents, their definition of the multi-parent operator is to define the cross operator with n string division points. Overall, scheduling problems with a multi-objective nature may be solved well using meta-heuristic search optimization techniques. However, the bulk of methods for resolving issues with a limited number of objectives employed algorithms. Given that changing the PAS problem to a mulz3ti-objective problem entails a large number of objectives derived from soft constraints, in order to ensure convergence behavior, the addition of a large number of objectives necessitates particular adjustment to the searching criteria. Aside from that We can observe that the scheduling programme made use of a meta-heuristic multi-objective optimization approach that included particle and genetic based searches. Additionally, the bulk of them require special operator designs depending on the application's nature and cannot be used directly. Table 2 lists all of the papers that addressed the PAS/NRP dilemma. It is observed from Table 1 that the literature contains many multi-objective metaheuristic algorithms, however, all of them have dealt with the multi-objective as single objective based on weighted average of the objectives which subject to local minima. To handle this, it is needed to propose non-dominated sorting based multi-objective optimization. On the other side, we observe from Table 2 that the number of soft-constraints ranges between 5 to 10 which makes the problem as candidate many objective optimizations instead of traditional multi-objective optimization when we consider the soft-constraints as objectives of the problem.

Table 2: Pseudocode of the process of selecting non-dominated solutions based on the process of NSGA-III.

```
Input:
- H structured reference points Zs or supplied aspiration
- points Za,
- parent population Pt
Output:
- P(t+1)
Start
1: St=∅,i=1
2: Qt = Recombination+Mutation(Pt)
3: Rt = Pt ∪ Qt
4: (F1,F2,...)=Non-dominated-sort(Rt)
5: repeat
6:    (St = St ∪ Fi and i = i + 1
7: until|St|≥N)
8: Last front to be included: Fl = Fi
9: if|St|=N    then
10:    P(t+1) = St, break;
11: else
12:    P = all previous fronts
13:    Points to be chosen from Fl: K = N − |Pt+1|
14:    Normalize objectives and create reference set Zr:
   Normalize(fn,St,Zr,Zs,Za)
15:    Associate each member s of St with a reference point:
[π(s),d(s)] =Associate(St,Zr) % π(s): closest reference point, d:
   distance between s and π(s)
16:   Compute niche count of reference point
17:   Choose K members one at a time from Fl to construct P(t+1):
   Niching(K, ρj, π, d, Zr, Fl, P(t+1))
18: End If
End
```

Table 3: Review of articles worked on solving PAS problem

Author	**Application**	**Hard constraints**	**Soft constraints**	**Optimization method**	**Type**
Demester [16]	PBAS	8	5	Hybrid Tabu search with heuristics	Static
Sara [33]	PBAS	2	10	Tabu local search	Dynamic
Saif [19]	PBAS	5	6	Adaptive deluge algorithm	Static

Table 2: Overview of multi-objective optimization in scheduling problems

Reference	Method/Technique	Key Features	Application	Limitations/Improvements
Ghasemi, Khalili-Damghani, et al. (2019)	MMOPSO	Mixed-integer model, focus on cost and supply deficit	Earthquake response	Superior to NSGA-II and epsilon constraint method
Adhikari and Srirama (2019)	Modified PSO	Optimizes energy use and computing time	IoT scheduling in cloud	Weighted sum approach for multi-objective handling
Yang, Deb et al. (2011)	APSO	Improved convergence through individual and global bests	General optimization	Reduces unpredictability, addresses speed and accuracy limits
Fang and Popole (2019)	Modified PSO with SOM	Neighborhood generation, neighborhood best solution selection	PSO performance enhancement	Provides only single solution
[23]	SAPSO & SAMOPSO	Self-adaptive PSO, external repository for Pareto front	Multi-objective optimization framework	Cyclic sorting, elitist-preserving principles
[24]	Modified PSO	Addresses large dimensional discrete variables	General optimization	Uses stretching, neighborhood search techniques
[25]	Modified NSGA-II	Resource allocation in operating room scheduling	Surgery scheduling	Focuses on allocating ORs to surgical specialties
[26]	Multi-parent crossover genetic algorithm	Multi-parent operator for n parents	Genetic algorithm variation	Does not have non-domination sorting perspective

4 Research gap

It is observed that in the domain of Patient Admission Scheduling (PAS) and similar scheduling challenges, most studies predominantly utilize techniques that manage multiple objectives through a weighted average approach. While this method is widely accepted, it is often prone to leading to local minima, thereby potentially yielding suboptimal solutions.

Furthermore, the literature demonstrates a significant absence of non-dominated sorting approaches in multi-objective optimization for scheduling problems. Non-dominated sorting plays a crucial role in identifying truly optimal solutions across a range of objectives, without unfairly favoring any single one. This aspect of optimization is particularly important in scenarios where a balanced consideration of multiple factors is essential.

Additionally, the current methodologies in the field largely concentrate on traditional multi-objective optimization. However, in scenarios such as PAS, where the number of soft constraints is considerable, ranging between 5 to 10, the issue becomes more aligned with many-objective optimization. This transition from multi-objective to many-objective optimization is not sufficiently addressed in the existing research, indicating a gap in the approach to handling complex scheduling problems with a multitude of objectives.

5 Methodology

This section presents the developed methodology for our dynamic patient's admission scheduling. It starts with presenting the pre-processing in sub-section 6.1. Next, the window- based NSGA-III in sub-section 6.2. Next, we present the selection of confirmed and non-confirmed patients in sub-section 6.3. Afterwards, the variable length optimization of window- based NSGA-III is given in sub-section 6.4. Lastly, the evaluation metrics are provided in sub-section 6.5.

1.1 Problem formulation

Assuming that we have a hospital combined of set of departments D under various specialisms S and each department contains set of rooms under the department R. In addition, we assume that we have an arrival rate of patients to the hospital where each patient requires serving it within certain number of nights inside a preferred department and by type of specialism. In addition, each room has certain capacity for accommodating pre-defined number patients at once. Our problem is about allocating the patients inside the rooms within period of time (number of nights) using solution vector x with minimizing the violation of soft-constraints $(f_1, f_2, \dots f_n)$ and preventing the violation of hard-constraint $(h_1, h_2, \dots h_m, g_1, g_2, \dots g_k)$.

The solution is combined of set of components that defines the allocation of each patient at each night for the selected room. In other words, the solutions length equals to the number of patients, and each component inside the solutions is a tuple of tree values, namely, the index of the bed that is assigned to the patient, the starting night, and the ending night. This problem is formulated as multi-objective optimization problem as:

$$x = argmin\ (f_1, f_2, \dots f_n) \quad (1)$$

$$s.t.\ \ g_1 = 0, g_2 = 0, \dots, g_k = 0$$
$$h_1 \geq 0\ , h_2 \geq 0, \dots h_m \geq 0$$

Hence, the problem is formulated mathematically as multi-objective optimization problem with many objective functions, many hard and soft-constraint. According to [17], this is regarded as NP-hard problem.

Assuming that the outcome of the optimization after running at time t it is PF_t. We use the penalties of the soft-constraint to provide ranking of the solutions based on the overall cost in a descending manner. This is done using this Equation (2)

$$y_j = \sum_{i=1}^{NSC} w_i f_i(x_j) \quad (2)$$

Where:

x_j is a solution selected from the Pareto Front

w_i is the penalty that is associated with the soft-constraint i

y_j is the overall cost of the solution x_j

Next, we select the solution that has the lowest cost as the activated solution. From the activated solution, the algorithm selects the patients that are scheduled within three days as confirmed patients and the patients that are scheduled later than three days as non-confirmed patients.

The optimization problem is repeated in different days with different number of patients. The changing of the number of patients implies changing the length of the solution space. The algorithm will work on allocating selected patients of the non-confirmed list of patients.

1.2 Simulator

The simulator is presented in Figure 2. The newly arrived patients are fed into the scheduler which is responsible on receiving a solution from the solution selection block, and providing it to the list of non-confirmed patients. The list of non-confirmed patients provides its non-confirmed patients to a new call of the optimization algorithm and provides the patients that have their scheduled day within less than D days to the confirmed patients list through sub-block named confirm. The optimization algorithm operates on different lengths of solutions because of the change number of patients, consequently, the algorithm is named as variable length non-dominated sorting genetic algorithm.

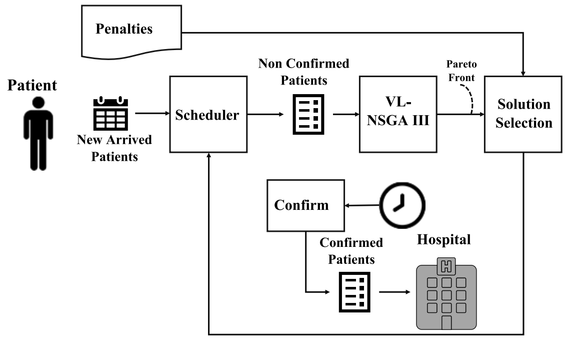


Figure 2: Simulation of dynamic patients scheduling using multi-objective optimization.

The following assumptions are inherent in the simulation model for the dynamic scheduling of patients in a hospital environment:

1. Hospital Structure: The hospital is composed of a set of departments D each specializing in various fields S, and containing a set of rooms R.
2. Room Capacity: Each room within a department can accommodate a pre-defined number of patients simultaneously.
3. Patient Arrival Rate: There is a specific rate at which patients arrive at the hospital, and each patient requires a certain number of nights within a preferred department and specialization.
4. Service Duration: Each patient is to be served within a specified number of nights.
5. Dynamic Solution Space: The optimization problem is dynamic, with the solution space changing in length due to the varying number of patients on different days, affecting the allocation of patients from the non-confirmed list.
6. Time-Dependent Optimization Outcome: The outcome of the optimization process at time t is denoted as PF_t indicating a time-dependent Pareto Front.

1.3 General algorithm

The algorithm of the scheduling combines the optimization with additional steps in order to enable dynamic scheduling. Firstly, there is a pre-processing step with the goal of preparing prior calculation of the various soft-constraints values. This enables shorter execution time of the optimization throughout the time interval of scheduling. Secondly, the new arrived patients are entered to queue according to their arriving time and the queue has a certain length so when the queue if full again the optimization is conducted and the new patients are located and the non-confirmed patients are allowed to be re-located. Thirdly, an algorithm for selecting one solution from the pareto front is enabled after running the optimization. This algorithm uses a weighted average formula of the soft-constraint according to a penalty entered from the user. Fourthly, the solution is activated and patients from the queue are decomposed into two sets: the first one is the confirmed patients and the second one is the non-confirmed patients. The difference between the two sets is that the confirmed patients are the patients that are scheduled with three days from the current date while the non-confirmed patients are the patients that are scheduled later than three days as long as their scheduling does not exceed the permitted period. A pseudocode of the general algorithm is given in Table 4.

Table 4: Pseudocode of the general scheduling algorithm using queue, multi-objective optimization and solution selection algorithm.

```
Input:
- w: Weights of the soft-constraints penalties
- Q: Queue used for storing new patients before re-running the
  MOO optimization
- timeInterval: Time interval for scheduling
- It: Number of iterations for the MOO optimization
- popSize: Size of the population in the optimization
- Rooms: Room matrix with information about supported
  departments, specialisms, and capacities
Output:
- schDecision: Scheduling decision, assigning each patient to a
  room
Start:
1: Pre-calculate soft-constraints using preProcessing (Rooms, w)
2: For each time interval in timeInterval
3:    While Q is not full
4:       Add new patient to Q
5:    End while
6:       Run MOO optimization using Optimization (popSize, It)
7:       Select solution using selectSolution (paretoFront, w, soft-
  constraints)
8:        Divide patients into confirmed and non-confirmed using
  assignFrom (Q, solution)
9:        Remove confirmed patients from Q and add them to
  schDecision
10:        Add non-confirmed patients to Q
11: End for
12: Return schDecision
End
```

1.4 Pre-processing

The goal of the pre-processing is to execute pre-calculation of the possible values of soft-constraints penalties in advanced according to all possible values of violations. As an example, For the gender constraint violation, assuming that we have n patients inside a room, it is possible to have mixed gender violation. This violation takes certain value if the majority are female and different value if the majority are males. Another example is the violation of the room capacity constraint, which takes different value according to the number of patients that exceed the room capacity. Assuming that the set of patients is denoted as $P = \{p_i\}, i = 1, \dots n$ and the set of rooms is denoted as $R = \{r_j\}, j = 1, \dots m$ where $n \gg m$. However, the patients arrive based on an arrival rate λ. Instead of calculating the soft-constraint based on the patient using function $f(p_i, r_j)$, we map the patient to a class or category according to his gender, needs or preference $C_p(p_i)$, and the room to a class or category according to its occupied patients, department and supported specialism $C_r(r_i)$ and we apply pre-calculated function for providing the soft-constraint or violation $f(C_p(p_i), C_r(r_i))$. Considering that the number of values of $C_p(p_i)$ and $C_r(r_i)$is limited then the generating the of the corresponding soft-constraint is more efficient by using $f(C_p(p_i), C_r(r_i))$ instead of $f(p_i, r_j)$.

1.5 Initialization algorithm

The initialization algorithm is in charge of creating the primary arrangement interior the window, which signifies the number of days which will handle a specific number of unused understanding candidates. S_pre, which stands for the arrangement decided based on the past window, and Information, which stands for the information that comprises numerous sorts of data, essentially a list of rooms, an overhauled list of patients, and the fittingness of the patients for the rooms, are the inputs for this strategy. The arrangement after optimization based on the current window and upgraded persistent list is demonstrated by the yield, S current. The strategy cycles through the List-new-patients and begins a variable called Room with the esteem of -1, showing that a appropriate room has not however been found for this quiet. A deferred persistent or a patient who wasn't deferred is the persistent in address. Within the previous situation, it decides whether or not the room from the earlier arrangement is suitable by checking it. The quiet is put in this room since it is appropriate and open. Something else, in case there are any open rooms, a irregular room is chosen for this persistent. The understanding is designated to his room from a earlier arrangement or at arbitrary within the occasion that no open rooms are accessible, and it receives a delay, giving the hail delay a esteem of 1.

Table 5: The generation of the initial solution.

Input
- S_{pre} // *previous solution*
- $Data$ //*includes rooms and patients and Room-Patient-Suitability*
- W // *the current window of performing the new optimization*

Output
- $S_{current}$ // *initial solution for current window*

Start Algorithm
for patient in List of patients from solution
    Room ← -1 //*initialization*
    if the patient is delayed (not new)
        if initial room still has space **AND** this room is suited for this patient
            $Room \leftarrow previous\ day\ solution\ room$
        end if
    end if
    while not (Room is suited and has space) and there is more Rooms
        $Room \leftarrow random\ (Rooms)$
    end while
    if Room not equal to -1
        Assign patient to Room.
        Set his delay value to zero.
    else // *the case the room is still* -1
        assign patient to Room // *if it's delayed, we can use a not suited room.*
        set his delay value to one.
    end if
end for
End Algorithm

1.6 Crossover

Crossover's function is to create a new generation from an existing one, which promotes exploitation, while mutation's function is to tweak an existing solution in some way, which promotes exploration. In genetics, both crossover and mutation exist. The algorithm for the crossover is shown in Table 6. The input consists of the entire population and IN, which denotes the proportion of the population where crossover is carried out. The elites, who stand for the generation's best answers, are typically subject to the crossover.

The population after crossover is the output. The algorithm chooses two random crossover solutions for each crossover iteration and creates a random fraction of patients to shift their rooms and assign them to DeltaRooms from each crossover solution. Additionally, it creates a random sample of patients and sends them to DeltaDelay in order to adjust their delay. Then it makes the necessary changes to the initially chosen two parents and includes the off-springs in the new generation.

Table 6: The crossover operation for the genetic design.

Input:
- current generation,

Output:
- new generation

Start Algorithm
Choose a random portion of the generation to apply crossover to.
for counter IN portion size
    Choose two parents x,y from the current generation
    DeltaRooms ← random portion of patients to change their rooms from solution x to solution y.
    DeltaDelay ← random portion of patients to change their delay from solution x to solution y.
    Child 1=change (x, y, DeltaRooms, DeltaDelay)
    Child 2=change (y, x, DeltaRooms, DeltaDelay)
    Add child 1 and child 2 to new the generation
end for.
End Algorithm

1.7 Mutation

For the mutation, the pseudocode is presented in Table 7. The input of the algorithm is the individual or solution that will be selected for mutation, the mutation rate which indicates to how many patients in the Individual receptivity to change and acceptance rate ap determine whether or not we adopt the dominating solution following mutation. This step is taken to make it possible to avoid local minima.

After mutation, the output is altered individually. As can be seen from the pseudocode, the algorithm chooses at random either the type 1 or type 2 neighborhood type before performing the mutation on the chosen person. The algorithm then verifies domination and accepts the solution if it is the dominant one. It accepts non-dominance with a probability known as the acceptance rate. The objective is to make the algorithm more explorable.

Table 7: The mutation operation for the genetic design.

Input:
- Solution
- Mutation rate: how many patients in the individual to change.
- ap: acceptance rate

Output:
- new Solution with mutated individuals

Start Algorithm
select random neighborhood
$new - Solution \leftarrow neighborhood$ (Solution, Mutation rate)
If new- Solution Dominates the current Solution

```
    current Solution ← new − Solution
Else
Generate a probability to allow bad Solutions
if generated probability > ap
    current Solution ← new- Solution
End for
End Algorithm
```

Neighborhood 1 or Neighborhood 2—shown in Tables 8 and 9 respectively—are the bases for the neighborhood operation. While neighborhood 2 focuses on changing the delay of random patients randomly, neighborhood 1 focuses on changing the location or room of random patients at random. In order to provide the searching method more latitude, both of them must be employed in the mutation.

In Table 8, the mutation rate and the current solution.

Table 8: Pseudocode of neighborhood 1 operator used in the mutation.

```
Input:
- Mutation rate
- Current Solution
Output:
- new Solution after the change
Start Algorithm
While Mutation rate
    patient ← random (current Solution patients)
    new-room ← random (current Solution rooms)
    if the new-room is suited for this patient
        set the patients room to the new-room.
    end if
end while
End Algorithm
```

Table 9: Pseudocode of neighborhood 2 operator used in the mutation.

```
Input:
- Mutation rate
- Current Solution
- Window
Output:
- new Solution after the change
Start Algorithm
while Mutation rate
    patient ← random (current individual patients)
    new-delay ← random (1 ← 0)
    if the new-delay + day is in the patients staying range
        set the patients delay to the new-delay.
    end if
end while
End Algorithm
```

1.8 Solution sorting

For sorting solutions, we use domination operators. The only domination operator is non-dominated sorting which has the role of sorting the solutions into ranks, the first rank includes the non-dominated solutions over the entire population. The second rank includes the solutions that are dominated by the first rank and dominating other ranks and so on. The algorithm is divided into a main. The algorithm of solutions ranking is tasked with orchestrating the entire sorting process, where fronts are initialized, and each solution in the population is systematically evaluated and ranked. The algorithm commences by initializing separate fronts, each intended to group solutions of equivalent non-domination levels. The core of the algorithm involves a thorough evaluation of each solution in the population to determine its dominance relationships. Solutions are compared pairwise, leading to the identification of those dominated by and dominating each solution. The first front is populated with solutions that are not dominated by any other, representing the optimal trade-offs. Subsequent fronts are iteratively constructed, where each front consists of solutions only dominated by those in the preceding front. This iterative process continues until all solutions are assigned to a rank, effectively segregating the population into distinct layers of non-dominated sets. The outcome is a hierarchically structured set of solutions, providing a clear perspective on their relative quality and guiding the selection process in the evolutionary algorithm.

Table 10: Pseudocode of solutions ranking

```
Inputs:
• Population P: A set of N solutions.
Outputs:
• Ranked Fronts: Sets of solutions sorted into different ranks based on
  non-domination.
Start Algorithm
1. Initialize Fronts:
   Create empty lists for each front (Front 1, Front 2, ...).
2. Evaluate and Rank Each Solution:
   for each solution p in Population P:
     Initialize dominatedByP (list of solutions dominated by p) as an
     empty list.
     Initialize dominatesP (count of solutions that dominate p) as zero.
     for each solution q in Population P:
       if p dominates q, add q to dominatedByP.
         if q dominates p, increment dominatesP.
           if dominatesP is zero (i.e., p is not dominated by any other
           solution):
             Assign p to Front 1.
3. Construct Subsequent Fronts:
     Initialize Current Front as Front 1.
     while Current Front is not empty:
       Initialize Next Front as an empty list.
       for each solution p in Current Front:
         for each solution q in dominatedByP of p:
           Decrement dominatesP counter for q.
           if dominatesP for q becomes zero:
             Assign q to Next Front.
             Replace Current Front with Next Front.
4. Return the Ranked Fronts:
    The fronts are ranked such that Front 1 contains solutions not
    dominated by any other, and each subsequent front contains
    solutions only dominated by those in the previous front.
End Algorithm
```

procedure and two sub-procedures, each fulfilling distinct roles

1.9 Selection of solution

The result of the optimization when it is applied is a Pareto front which represents set of non-dominated solutions. Thus, we need an algorithm that selects solution out of the Pareto front for enabling it in the scheduling. Assuming that the weights of the soft-constraints or the objectives are represented by a vector $w = [w_1\ w_2\ \dots w_m]$ where $w_1 + w_2 \dots + w_m = 1$. The solutions will be ranked based on linear production between the weights and the values of the objective function. In other words, each solution x_i from the pareto front will be mapped to one cost value based on the Equation (3)

$$x = argmin\ (f_1, f_2, \dots f_n) \quad (3)$$

$$f(x_i) = w{y_i}^T$$

where

$$w = [w_1\ w_2 \ldots w_m]$$

$$y = \begin{bmatrix} y_{i,1} \\ y_{i,2} \\ \\ \\ y_{i,m} \end{bmatrix}$$

After that, the solutions are sorted in an ascending manner according to the cost values or $f(x_i)$ and the first solution or the solution that has the least cost value is selected and enabled. The result of enabling the solutions is two set of patients: the first one is confirmed set S_{conf} and it includes patients that are scheduled within three days and the second one is the non-confirmed set $S_{non-conf}$ and it includes patients that are scheduled later than three days. For S_{conf}, we remove them from the queue so they will not be used again for re-scheduling while for $S_{non-conf}$ we keep them in the queue so they are allowed for rescheduling in the next execution of the algorithm.

1.10 Variable length optimization of Window Based NSGA-III

In order to distinguish between patients that are allowed for rescheduling from new arrived patients, we use variable length optimization (VLO). In VLO, different lengths of solutions are used where each solution allow for rescheduling of different sub-sets of the non-confirmed patients. The goal of this is to conduct optimization with giving more importance to rescheduling of later scheduled patients and less importance of earlier scheduling patients.

The optimization in this case, will generate different number of solutions according to the number of patients where the solutions that contains earlier scheduled patients are less than the solutions of later scheduled patients. We call this algorithm variable length NSGA-III or VL-NSGA-III.

1.11 Evaluation metrics

The evaluation metrics that were employed to assess our created strategy are provided in this subsection. It has broken down.

- **Set coverage:**

This metric compares the Pareto sets P_{s1} and P_{s2} as follows

$$c(P_{s1}, P_{s2}) = \frac{|\{y \in P_{s2} \mid \exists x \in P_{s1}: x > y\}|}{|P_{s2}|} \quad (4)$$

C is equal to the number of solutions in Ps2 divided by the proportion of non-dominated solutions in Ps2 that are dominated by non-dominated solutions in P s1. Therefore, it is crucial to reduce the value of C (X, P s) for all pareto sets X while assessing a set Ps.

- **Hyper-Volume**

The HV-metric has been used widely in evolutionary multi- objective optimization to evaluate the performance of search algorithms. It computes the volume of the dominated portion of the objective space relative to a worst solution (reference point); this region is the union of the hypercube whose diagonal is the distance between the reference point and a solution x from the Pareto set PS. Higher values of this measure indicates to more desirable solutions. HV is given by the Equation (5).

$$HV = \text{volume}\left(\cup_{x \in P_s} \text{HyperCube}(x)\right) \quad \textbf{(5)}$$

6 Experimental works And evaluation

The assessment is a simulator-based assessment. For this stage, we utilized the simulator's data, which covered a total of 36 days. The data has similar layout to the data provided in the work of [34]. We contrasted NSGA-3, which incorporates numerous objective optimizations based on our created operators, with the following benchmarks: particle swarm optimization (PSO), multi-objective particle swarm optimization (MOPSO), and objective decomposition particle swarm optimization (ODPSO). The set coverage, hyper-volume, and convergence curves were produced.

1.12 Set-Coverage

The results of the set-coverage reveal the superiority of NSGA-III over the benchmarks. More specifically, NSGA-III has accomplished full domination over PSO which is single optimization algorithm, full domination over both MOPSO and ODPSO which are multi-objective algorithms, and 0.66 domination over NSGA-II. On the other side, non-of the algorithms of ODPSO, MOPSO, and PSO were capable of dominating NSGA-III. However, NSGA-II has provided 0.96 percentage of domination over NSGA-II.

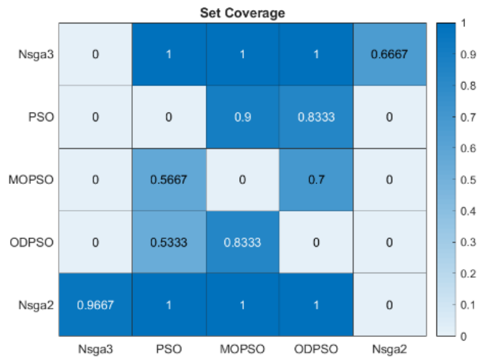


Figure 3: Set coverage of our developed WB approach and it is comparison with the benchmarks.

Hyper-volume

The results of the hyper-volume are presented in Figure 4. We find that the hyper-volume generated from NSAG-III and NSGA-II were the highest compared with the other approaches

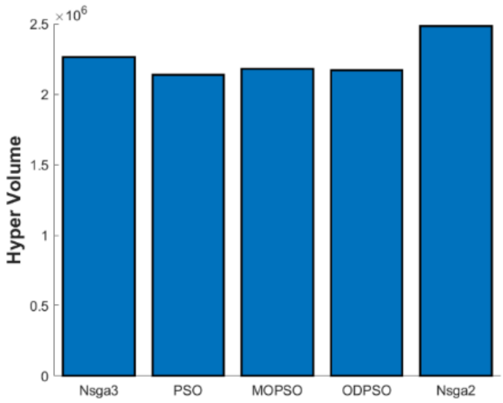
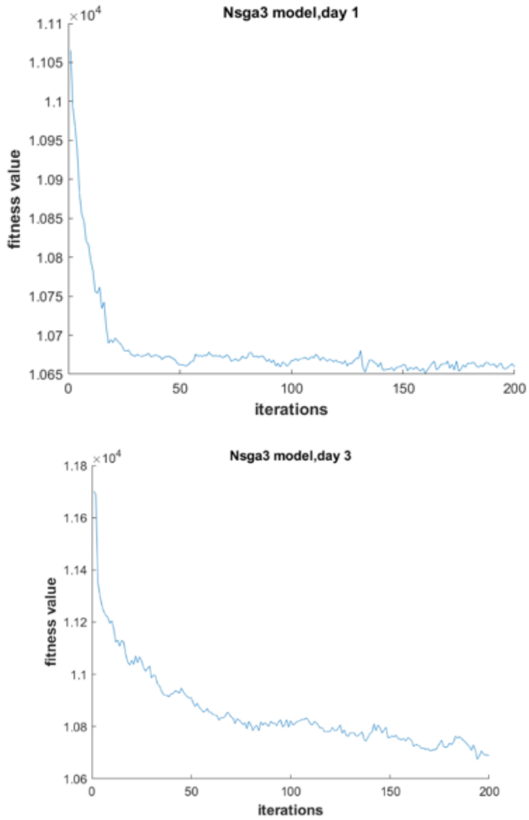


Figure 4: Hyper-volume of our developed algorithm and its comparison with the benchmarks.

1.13 Convergence curve

Considering that the optimization is reapplied in every day, the convergence curve is plotted to show the effectiveness of the optimization. The convergence curve is plotted based on fitness value equals to the average of the objectives. For plotting the convergence curve, we use calculate a fitness value as weighted average of the soft constraints based on the penalties of them. In Figure 5, we present the convergence of days 1, 2, 3 and the last day 36.



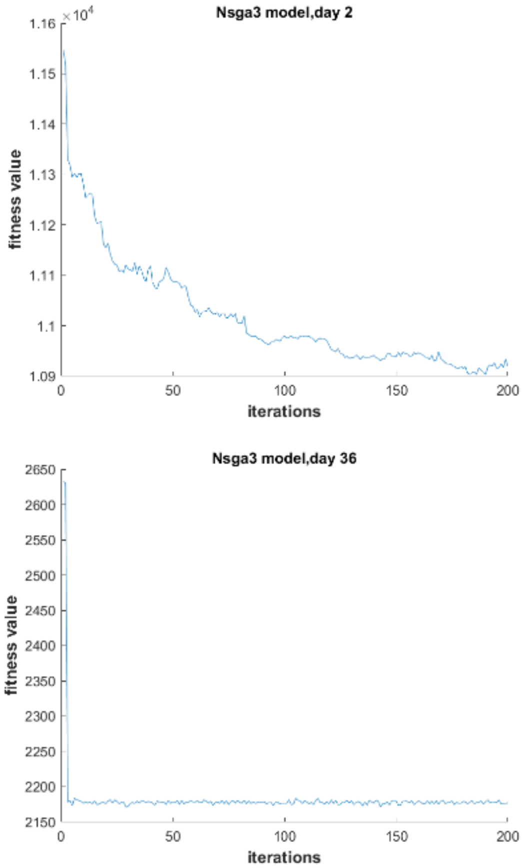
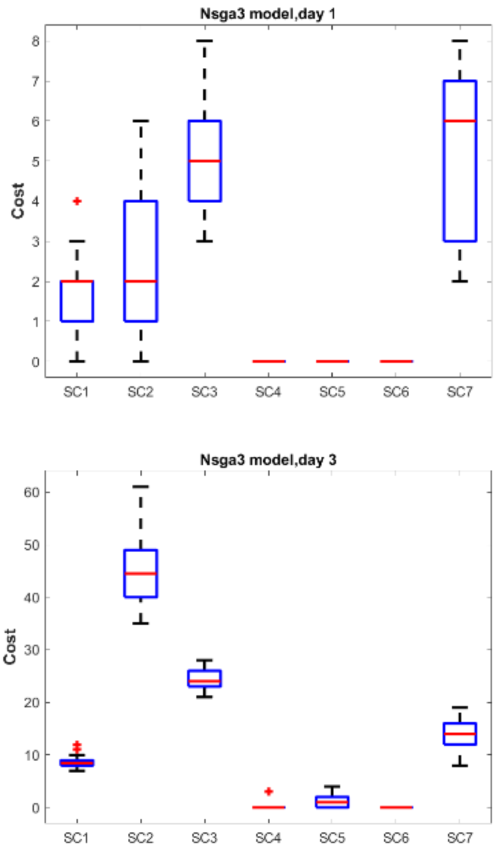


Figure 5: The convergence curve of NSGA-III of some of the optimization days.



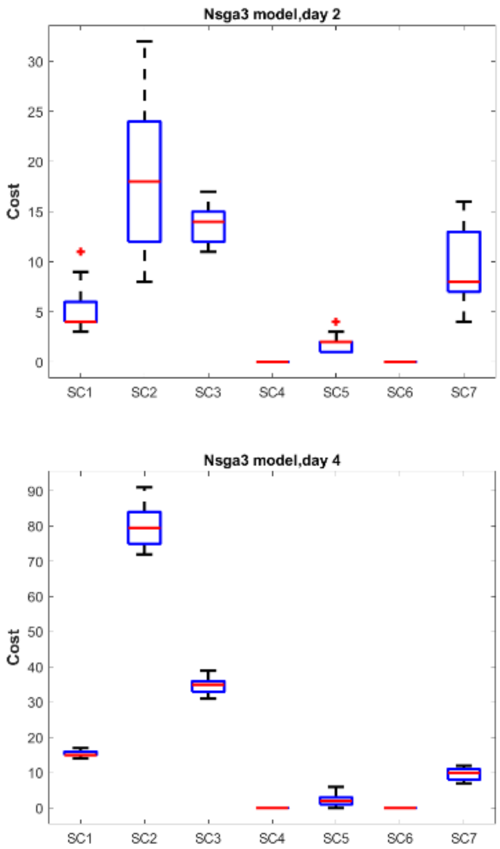


Figure 6:The boxplot of soft-constraints of NSGA-III of some of the optimization days.

1.14 Soft-constraints-values

In addition to the set-coverage, hyper-volume and convergence curve, we present the soft-constraints of each day Pareto front as boxplot diagram in Figure 6. The soft-constraints are encoded according to the symbols provided in Table 10.

Table 11: Coding for the soft-constraints used in the optimization.

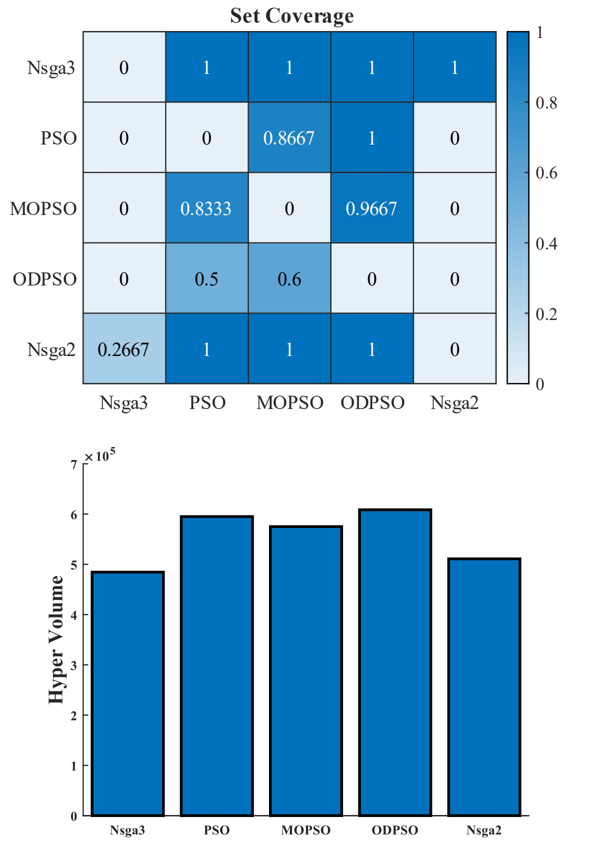
Code	Meaning
SC1	Missing Room Equipment
SC2	Unsatisfied Room Preference
SC3	Partial Specialty Level
SC4	Unsatisfied Gender Policy
SC5	Over -Crowd Risk
SC6	Delay
SC7	Transfer

The visualization shows a similar performance between the various days in the relative relation between the soft-constraints with changing in the values obtained from one day to another.

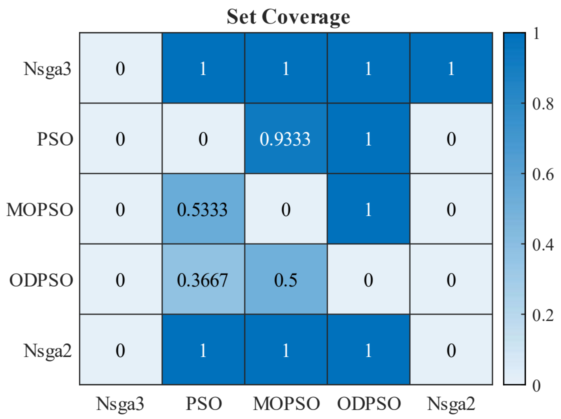
This is interpreted by the effect of the dynamic in the performance that changes from one day to another. However, associating this graph with the convergence graph given earlier shows that the algorithm was capable of handling the dynamics and brining the cost to a lower value.

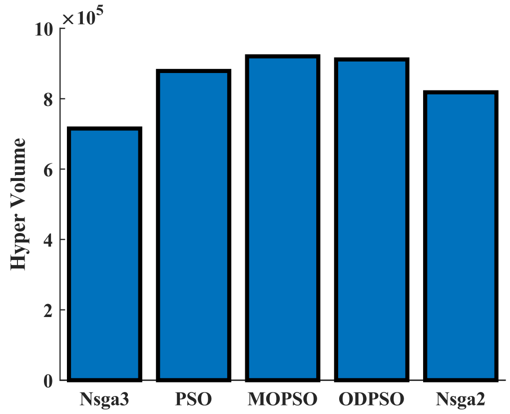
1.15 Robustness evaluation scenarios

For evaluating our algorithm more comprehensively, we conducted a robustness evaluation by increasing the arrival rate of patients in the range of 15, 20, 25, and 30 patients per day. For each scenario, we generated the values of set coverage and hyper-volume. Observing the results of the set coverage as depicted in figure – confirms our finding of the superiority of of NSGA-III over other benchmarks. This is concluded from the domination of NSGA-III compared with the other optimization algorithms. It is found that a full domination was obtained when the arrival rate was 15. This is associated with high values of hyper-volume and competitive to other methods. Hence, it is found that increasing the arrival rates of patients has not only maintained the superiority but also the diversity of decision making.

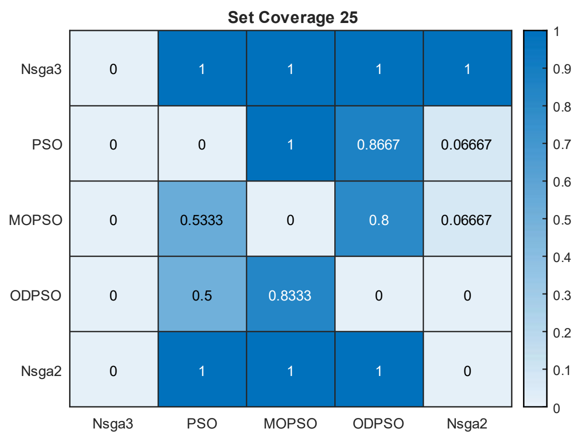


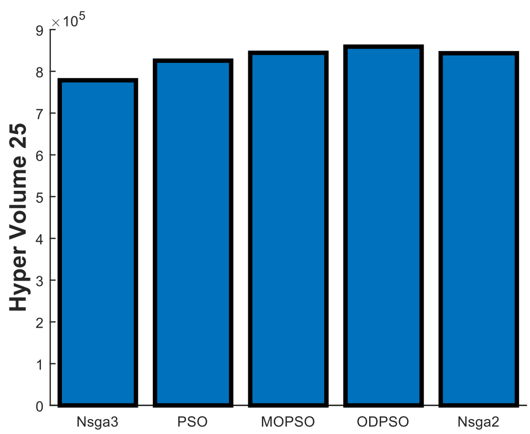
15



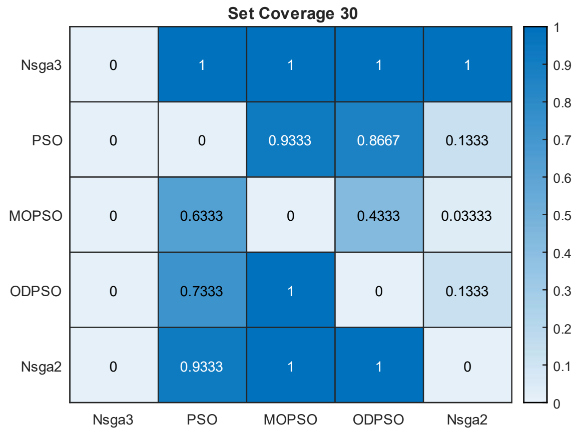


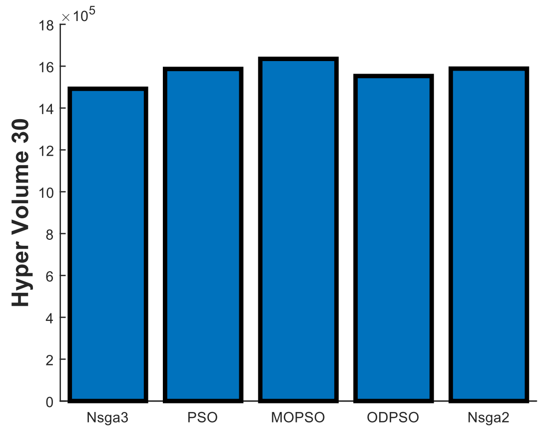
20





25





30

Figure 7:Set coverage and hyper-volume for different values of arrival rates ranging from 15 until 30

7 Conclusion and future work

Dynamic patient scheduling for hospital admission is challenging combinatorial problem with dynamical nature and many soft-constraints. An effective approach for solving it is using many-objective optimization MOO algorithms. However, direct application of them is not feasible due to the static nature of MOO algorithms. Hence, handling this application requires incorporation of other assisting blocks.

In this article, we have developed a novel simulator for dynamic scheduling of patients with window and coordinator. The role of the window is to accumulate both newly arrived patients and non-o patients.

The coordinator's duties include choosing a subset of patients from the window, placing them in the optimization block on one side, and choosing a non-dominated solution, activating it in the hospital on the other. A rigorous 36-day evaluation using PSO, ODPSO, MOPSO, NSGA-II, and NSGA-III has shown that NSGA-III is superior based on set-coverage and soft-constraints.

The practical implications of the findings from this proposed solution have been deemed to hold significant promise for enhancing the efficiency of hospitals and healthcare systems. Improved resource utilization, reduced patient wait times, and elevated overall care quality could be achieved through the implementation of a dynamic scheduling system based on multi-objective optimization. Despite these benefits, challenges such as the integration with existing healthcare systems, staff training, and the need for robust data privacy and security measures have been identified as potential obstacles. Furthermore, the scalability and customization required for the system to be successfully adopted across various healthcare settings present additional complexities. A gradual, phased approach to implementation, involving pilot testing and stakeholder engagement, can be suggested to mitigate these challenges and facilitate smoother adoption.

Future research is to explore the adaptability of the methodology used in the healthcare scheduling system to other complex scheduling problems across different domains. The manufacturing sector, transportation and logistics, energy management, education, event management, and urban planning have been identified as areas where similar optimization techniques could be applied. Each domain presents its unique set of challenges and constraints, necessitating the customization of the optimization framework. The extension of this research into varied domains is expected to account for specific requirements and challenges while considering the effects on human behavior, regulatory standards, and economic considerations.

References

[1] I. Papanicolas, L. R. Woskie, and A. K. Jha, "Health care spending in the United States and other high-income countries," *Jama*, vol. 319, no. 10, pp. 1024-1039, 2018.

[2] N. Fares, R. S. Sherratt, and I. H. Elhajj, "Directing and orienting ICT healthcare solutions to address the needs of the aging population," in *Healthcare*, 2021, vol. 9, no. 2, p. 147: MDPI.

[3] J. Meehan, L. Menzies, and R. Michaelides, "The long shadow of public policy; Barriers to a value-based approach in healthcare procurement," *Journal of Purchasing Supply Management*, vol. 23, no. 4, pp. 229-241, 2017.

[4] R. Guido, V. Solina, and D. Conforti, "Offline patient admission scheduling problems," in *International Conference on Optimization and Decision Science*, 2017, pp. 129-137: Springer.

[5] A. N. Mahmed and M. Kahar, "Window-Based Multi-Objective Optimization for Dynamic Patient Scheduling with Problem-Specific Operators," *Computers*, vol. 11, no. 5, p. 63, 2022.

[6] C. Taramasco, B. Crawford, R. Soto, E. M. Cortés-Toro, and R. Olivares, "A new metaheuristic based on vapor-liquid equilibrium for solving a new patient bed assignment problem," *Expert Systems with Applications*, vol. 158, p. 113506, 2020.

[7] R. Guido, M. C. Groccia, and D. Conforti, "An efficient matheuristic for offline patient-to-bed assignment problems," *European Journal of Operational Research*, vol. 268, no. 2, pp. 486-503, 2018.

[8] K. Hussain, M. N. M. Salleh, S. Cheng, and Y. Shi, "Metaheuristic research: a comprehensive survey," *Artificial Intelligence Review*, vol. 52, no. 4, pp. 2191-2233, 2019.

[9] R. Alizadeh, J. Rezaeian, M. Abedi, and R. Chiong, "A modified genetic algorithm for non-emergency outpatient appointment scheduling with highly demanded medical services considering patient priorities," *Computers Industrial Engineering*, vol. 139, p. 106106, 2020.

[10] K. Dorgham, I. Nouaouri, H. Ben-Romdhane, and S. Krichen, "A hybrid simulated annealing approach for the patient bed assignment problem," *Procedia Computer Science*, vol. 159, pp. 408-417, 2019.

[11] A. Hammouri, "A modified biogeography-based optimization algorithm with guided bed selection mechanism for patient admission scheduling problems," *Journal of King Saud University-Computer Information Sciences*, 2020.

[12] J. Luo, Q. Liu, Y. Yang, X. Li, M.-r. Chen, and W. Cao, "An artificial bee colony algorithm for multi-objective optimisation," *Applied Soft Computing*, vol. 50, pp. 235-251, 2017.

[13] D. Wang, D. Tan, and L. Liu, "Particle swarm optimization algorithm: an overview," *Soft Computing*, vol. 22, no. 2, pp. 387-408, 2018.

[14] R. Tanabe and H. Ishibuchi, "An easy-to-use real-world multi-objective optimization problem suite," *Applied Soft Computing*, vol. 89, p. 106078, 2020.

[15] H. R. Maier, S. Razavi, Z. Kapelan, L. S. Matott, J. Kasprzyk, and B. A. Tolson, "Introductory overview: Optimization using evolutionary algorithms and other metaheuristics," *Environmental modelling software*, vol. 114, pp. 195-213, 2019.

[16] P. Demeester, W. Souffriau, P. De Causmaecker, and G. V. Berghe, "A hybrid tabu search algorithm for automatically assigning patients to beds," *Artificial Intelligence in Medicine*, vol. 48, no. 1, pp. 61-70, 2010.

[17] A. M. Turhan and B. Bilgen, "Mixed integer programming based heuristics for the Patient Admission Scheduling problem," *Computers Operations Research*, vol. 80, pp. 38-49, 2017.

[18] B. Bilgin, P. Demeester, M. Misir, W. Vancroonenburg, and G. V. Berghe, "One hyper-heuristic approach to two timetabling problems in health care," *Journal of Heuristics*, vol. 18, no. 3, pp. 401-434, 2012.

[19] S. Kifah and S. Abdullah, "An adaptive non-linear great deluge algorithm for the patient-admission problem," *Information Sciences*, vol. 295, pp. 573-585, 2015.

[20] Y.-H. Zhu, T. A. Toffolo, W. Vancroonenburg, and G. V. Berghe, "Compatibility of short and long term objectives for dynamic patient admission scheduling," *Computers Operations Research*, vol. 104, pp. 98-112, 2019.

[21] M. Rezaeiahari and M. T. Khasawneh, "Simulation optimization approach for patient scheduling at destination medical centers," *Expert Systems with Applications*, vol. 140, p. 112881, 2020.

[22] A. K. Abera, M. M. O'Reilly, M. Fackrell, B. R. Holland, and M. Heydar, "On the decision support model for the patient admission scheduling problem with random arrivals and departures: A solution approach," *Stochastic Models*, vol. 36, no. 2, pp. 312-336, 2020.

[23] S. Ceschia and A. Schaerf, "Dynamic patient admission scheduling with operating room constraints, flexible horizons, and patient delays," *Journal of Scheduling*, vol. 19, pp. 377-389, 2016.

[24] S. Ceschia and A. Schaerf, "Dynamic patient admission scheduling with operating room constraints, flexible horizons, and patient delays," *Journal of Scheduling*, vol. 19, no. 4, pp. 377-389, 2016.

[25] B. Bilgin, P. Demeester, M. Misir, W. Vancroonenburg, and G. V. Berghe, "One hyper-heuristic approach to two timetabling problems in health care," *Journal of Heuristics*, vol. 18, no. 3, pp. 401-434, 2012.

[26] S. Kifah and S. Abdullah, "An adaptive non-linear great deluge algorithm for the patient-admission problem," *Information Sciences*, vol. 295, pp. 573-585, 2015.

[27] Y.-H. Zhu, T. A. Toffolo, W. Vancroonenburg, and G. V. Berghe, "Compatibility of short and long term objectives for dynamic patient admission scheduling," *Computers Operations Research for Health Care*, vol. 104, pp. 98-112, 2019.

[28] M. Rezaeiahari and M. T. Khasawneh, "Simulation optimization approach for patient scheduling at destination medical centers," *Expert Systems with Applications*, vol. 140, p. 112881, 2020.

[29] B. Tang, Z. Zhu, H.-S. Shin, A. Tsourdos, and J. Luo, "A framework for multi-objective optimisation based on a new self-adaptive particle swarm optimisation algorithm," *Information Sciences*, vol. 420, pp. 364-385, 2017.

[30] C. Seren, "A hybrid jumping particle swarm optimization method for high dimensional unconstrained discrete problems," in *2011 IEEE Congress of Evolutionary Computation (CEC)*, 2011, pp. 1649-1656: IEEE.

[31] Q. Lu, X. Zhu, D. Wei, K. Bai, J. Gao, and R. Zhang, "Multi-phase and integrated multi-objective cyclic operating room scheduling based on an improved NSGA-II approach," *Symmetry*, vol. 11, no. 5, p. 599, 2019.

[32] A. Arram and M. Ayob, "A novel multi-parent order crossover in genetic algorithm for combinatorial optimization problems," *Computers Industrial Engineering*, vol. 133, pp. 267-274, 2019.

[33] S. Ceschia and A. Schaerf, "Local search and lower bounds for the patient admission scheduling problem," *Computers Operations Research for Health Care*, vol. 38, no. 10, pp. 1452-1463, 2011.

[34] S. Ceschia and A. Schaerf, "Modeling and solving the dynamic patient admission scheduling problem under uncertainty," *Artificial intelligence in medicine*, vol. 56, no. 3, pp. 199-205, 2012.