OmniaScience

Journal of Industrial Engineering and Management
JIEM, 2026 – 19(2): 443-461 – Online ISSN: 2013-0953 – Print ISSN: 2013-8423
https://doi.org/10.3926/jiem.9219

A Simulation-Based Genetic Algorithm Approach for Solving No-Wait Two Stages Elective Surgery Scheduling Problem Under Uncertainty

Lawrence Al-Fandi[1*], Rahaf Abedalrohman[1], Najat Almasarwah[2]

[1]*Department of Industrial Engineering, Jordan University of Science and Technology, Irbid (Jordan)*
[2]*Department of Industrial Systems Engineering, Mutah University, Alkarak (Jordan)*

[*]*Corresponding author: lmalfandi@just.edu.jo*
riabedalrohman20@eng.just.edu.jo, Najat.eid@mutah.edu.jo

Received: November 2025
Accepted: June 2026

Abstract:

Purpose: The surgical unit is critical for hospital revenue. This study aims to address the no-wait two-stage elective surgery scheduling problem, focusing on minimizing makespan by considering operating room (OR) availability and eligibility, surgeon dedication, and recovery bed availability.

Design/methodology/approach: We propose a Mixed-Integer Linear Programming (MILP) model that integrates stochastic parameters for surgery duration and recovery time. A Genetic Algorithm (GA) is designed for deterministic conditions, while a simulation-based GA handles uncertainty.

Findings: Experimental results showed that GA provides near-optimal solutions for small instances within reasonable timeframes. At the same time, the simulation-based GA approach provided favorable outputs, with a relative gap to the lower bound of 6.531% on average and a CPU time of 56.286 seconds. The proposed model was applied to a real-world surgical suite, where results showed that the GA finds good solutions that deviate from the lower bound by an average of 4.053 % and takes 71 seconds of CPU time. Also, results showed that simulation-based GA provides solutions with makespan value significantly less than the actual scheduling in the hospital, where the makespan reductions range from 11.771 to 25.080 %.

Research limitations/implications: One of the limitations of this study is that it focused only on the surgery and recovery stage without considering the pre-operative stage. Also, the application of the proposed model was limited to small and medium-sized cases.

Practical implications: The proposed model and solution approach offer a practical, effective tool for hospital administrators to address the complexity of surgery scheduling, thereby enhancing operational efficiency in hospital settings.

Social implications: Improving the surgical scheduling process impacts the overall performance of the hospital by reducing overtime and waiting time, improving efficiency, increasing productivity, and thus reducing total costs.

Originality/value: This study addresses two-stage surgical scheduling while considering OR availability and eligibility, dedicated surgeons, Post-Anaesthesia Care Unit (PACU) bed availability, and two probabilistic parameters: surgery durations and recovery time. Also, using a simulation-based GA to solve the problem. These interactive aspects represent a research gap as they have not been comprehensively studied in previous research.

Keywords: elective surgery scheduling problem, operating room, recovery unit, makespan, genetic algorithm, simulation-based GA

To cite this article:

Al-Fandi, L., Abedalrohman, R., & Almasarwah, N. (2026). A simulation-based genetic algorithm approach for solving no-wait two stages elective surgery scheduling problem under uncertainty. *Journal of Industrial Engineering and Management*, *19*(2), 443–461. https://doi.org/10.3926/jiem.9219

1. Introduction

The healthcare sector and the efficient use of its available resources have become a major concern for various countries worldwide. These are due to the increasing demand for healthcare services, reduced budget, resource shortage, and pressure on the aging population (Xiao et al., 2016; Tayyab & Saif, 2022). The surgical unit is one of the most important and costly resources, generating more than 40% of the hospital's overall revenue (Denton et al., 2007). Planning and scheduling for surgical operations efficiently significantly affect the overall hospital performance (Farzad & Mohammad, 2016). In the surgery suite, the patient passes through three stages: pre-operative holding unit (PHU), intra-operative (operating room (OR)), and post-operative (Post-Anaesthesia Care Unit (PACU)) (Rahimi & Gandomi, 2021); each stage requires a different set of resources. Therefore, administrators must coordinate between these resources and schedule surgical operations efficiently, which is a very complex problem.

Different studies have been proposed recently to optimize the surgical scheduling problem subject to several performance measures, such as total cost, overtime cost, makespan, number of ORs used, overtime, surgeon satisfaction, etc., considering different constraints (Xue et al., 2025; Bektur & Aslan., 2024; Park et al., 2021; Hsu et al., 2021; Lin & Chou, 2019; Kamran et al., 2018). The eligibility of ORs is an important constraint that affects the OR scheduling problem and makes it complex, where ignoring this constraint leads to impractical scheduling decisions. Nevertheless, few studies covered OR eligibility (Wang et al., 2015; Wang et al., 2022; Tayyab et al., 2023), whereas most studies have studied and emphasized the importance of downstream facility management, such as PACU. However, planning and scheduling without integrating ORs with PACU lead to surgical cancellations, overtime, long patient waiting times, and deteriorating quality of care (Shehadeh & Padman, 2022).

Many researchers have addressed the surgical scheduling problem as a Hybrid Flow Shop (HFS) scheduling problem. Typically, the HFS scheduling problem combines classical flow shop and parallel machine environments (Fei et al., 2010). Some studies have focused on two stages of the HFS problem, where the first stage represents the PHU, and the second stage refers to the OR (Huang et al., 2012). Meanwhile, other studies considered the OR as the first stage, while the PACU as the second stage without wait constraint (Wang et al., 2015; Bam et al., 2017). On the other hand, some studies considered no wait constraint with inter-stage flexibility, which refers to the recovery process in the OR, or the recovery unit (Fei et al., 2010; Azaiez et al., 2021).

Most previous studies studied the surgical scheduling problem under deterministic conditions. However, in real conditions, many uncertain factors affect the scheduling process: patient arrival times, surgery durations, emergency cases, case cancellation, resource availability, and length of stay (Patrão et al., 2022). Taking uncertainty into account improves scheduling quality, making the scheduling process more realistic. Uncertainty in surgery duration and length of stay (LOS) in the recovery unit directly impacts scheduling efficiency. Over the years, many studies have focused on surgical scheduling alone under uncertainty with various uncertainty variables such as surgery duration and cleaning time duration (Clavel et al., 2020; Khaniyev et al., 2020). Other studies integrated surgical scheduling problems with the recovery units under uncertainty (Lin & Chong., 2025; Wang et al., 2020; Varmazyar et al., 2020).

Because the surgical scheduling problem is an NP-hard problem, and no exact solution can be found in a reasonable time. The approximate methods, such as heuristics and metaheuristics, could give near-optimal solutions in polynomial time (Rahimi & Gandomi, 2021). These approximate methods have been applied to tackle some real-world problems in the healthcare sector, such as scheduling the operations in the operating room (Abdeljaoued, et al., 2021; Aringhieri et al., 2015; Ramadan, et al., 2025), nurse rostering problem in the hospitals (Turhan & Bilgen, 2022), scheduling of radiotherapy treatment (Vieira et al., 2021), etc. Simulation-based optimization is also a powerful technique for dealing with complex problems under uncertainty in wide fields. It was first introduced in process system engineering by Subramanian et al. (2001), which combines a simulation with optimization techniques. Most studies in the surgical scheduling field assume that the surgery durations and recovery time follow the lognormal distribution (Bai et al., 2017; Zhang et al., 2020). However, other continuous distributions can be easily adopted using this approach. Some researchers used simulation-based optimization methods to treat surgical scheduling problems using discrete event simulation (Saremi et al., 2013; Gul et al., 2011), and others using Monte Carlo simulation (MCS) (Landa et al., 2016; Díaz-López et al., 2018).

This research will focus on proposing a mixed-integer linear program (MILP) for the no-wait HFS elective surgery scheduling problem. The model will consider dedicated surgeons, OR availability and eligibility in the first stage, and the PACU bed availability in the second stage. The model aims to minimize the makespan and includes two stochastic parameters: surgery duration time and the LOS in the recovery unit. Based on Table 1, these interactive aspects represent a research gap as they have not been comprehensively studied in previous research. The dedicated surgeon and OR eligibility constraints were previously considered in the same study but in a deterministic environment and without a recovery unit. Adding a recovery unit and including uncertainty led the problem to be more realistic. The simulation-based optimization approach will be used to solve this problem. Finally, a real-life case study will verify the proposed model's performance and the solution method's performance.

References	Decision level (operational)		Resource types							Operating rooms		Stochastic aspects		Criteria	Solution method
	Advanced	Allocation	PHU beds	ORs	PACU beds	Surgeons	Nurses	Anesthesiologists	Other	Identical	Non-identical	Surgery duration	Recovery time		
Dolatkhah et al. (2026)	✓	✓		✓						✓				Cost of opening ORs, surgery postponement, and overtime	Hybrid Reinforcement Learning and GA based Column Generation
Lin & Chong (2025)	✓	✓	✓	✓	✓					✓		✓	✓	Makespan	GA for Robust Scheduling
Swilugar & Herliansyah (2025)	✓	✓		✓	✓	✓	✓			✓				Total surgery completion time	GA
Xue et al. (2025)	✓	✓	✓	✓	✓	✓	✓	✓		✓				Total cost, average completion time and overtime	Honey Badger Algorithm based on Nash Equilibrium
Bektur & Aslan (2024)	✓	✓		✓		✓		✓		✓				Makespan	Artificial bee colony algorithm

References	Decision level (operational)		Resource types							Operating rooms		Stochastic aspects		Criteria	Solution method
	Advanced	Allocation	PHU beds	ORs	PACU beds	Surgeons	Nurses	Anesthesiologists	Other	Identical	Non-identical	Surgery duration	Recovery time		
Lin & Yen (2023)	✓	✓	✓	✓	✓					✓				Makespan	GA
Tayyab et al. (2023)	✓	✓		✓							✓			Overtime costs of ORs	Outer Approximation Method
Wang et al. (2022)	✓	✓		✓		✓					✓			Makespan	Adaptive composite dispatching method
Azaiez et al. (2021)	✓	✓		✓	✓					✓				Makespan	Heuristics
Hsu et al. (2021)		✓		✓				✓		✓				Makespan	Heuristic algorithms
Park et al. (2021)	✓	✓		✓		✓					✓			Number of ORs used, over time and Surgeon satisfaction based on preferred ORs	Surgery-based approach and the bundle-based approach
Wang et al. (2020)	✓	✓		✓	✓					✓		✓	✓	Completion time	Hybrid GA
Lin & Chou (2019)	✓			✓						✓				Utilization of the ORs, overtime-operating and wasting costs	Hybrid GA
Varmazyar et al. (2020)	✓	✓		✓	✓					✓		✓	✓	Makespan	Search algorithm embedding meta-heuristic and constructive heuristic
Díaz-López et al. (2018)	✓	✓		✓						✓		✓		Utilization rate of ORs	A simulation-optimization approach
Kamran et al. (2018)	✓			✓		✓				✓		✓		Tardiness, waiting time, over time, cancellation, and number Surgeon's surgery days	Sample average approximation and Bender's decomposition method

	Decision level (operational)		Resource types							Operating rooms		Stochastic aspects			
References	Advanced	Allocation	PHU beds	ORs	PACU beds	Surgeons	Nurses	Anesthesiologists	Other	Identical	Non-identical	Surgery duration	Recovery time	Criteria	Solution method
Bam et al. (2017)	✓	✓		✓	✓	✓				✓				Total costs	Heuristic
Wang et al. (2015)	✓	✓		✓		✓	✓	✓			✓			Makespan	Heuristic algorithms
Xiang et al. (2015)	✓	✓	✓	✓	✓	✓			✓	✓				Makespan	Ant Colony algorithm
Wang et al. (2014)	✓	✓		✓	✓	✓				✓				Total operating cost	Integrated the heuristic rules with a discrete particle swarm optimization algorithm
Fei et al. (2010)	✓	✓		✓	✓	✓				✓				Daily operating cost	Hybrid GA
Our study	✓	✓		✓	✓	✓					✓	✓	✓	Makespan	Simulation-based GA

Table 1. Summary of research studies

The simulation-based optimization approach used to solve this problem combines Monte Carlo simulation with a genetic algorithm. The GA was chosen because it is one of the most common methods used in studies to solve complex (NP-Hard) scheduling problems, providing near-optimal solutions within a reasonable computational time (Guzman et al., 2022). Furthermore, two stage surgical scheduling increases the complexity of the problem due to the interconnectedness and interdependence of the phases, making the GA a suitable and efficient option.

The reviewed research was compiled through a systematic search of major academic databases, including Scopus and Google Scholar. The search was based on relevant keywords such as operating room scheduling, surgical scheduling, and recovery bed allocation. A set of inclusion criteria was applied to ensure the relevance and quality of the selected studies. Only journal articles focusing on hospital resource scheduling and optimization problems at the operational level were included.

Table 1 focuses primarily on allocation and advanced optimization problems at the operational level within healthcare scheduling and operating room systems. Table 1 has been reorganized in ascending order based on publication year and author name to improve clarity, structure, and readability.

2. Materials and Methods

The research methodology was applied in five main steps, as depicted in Figure 1. Firstly, define and describe the problem, then propose a mathematical model that expresses the problem. The next step is solution method development, which includes building the GA and integrating it with MCS using Python. The third step is data collection preparation to be suitable for use in subsequent stages. It is an important stage to ensure the quality of the results, which includes data cleaning and data fitting. After that, several small and real instances are solved using a GA and simulation-based GA. Finally, methods validation by comparing solutions against lower bound (LB). The LB was utilized in literature to minimize the makespan in the two stages of HFS (Guinet et al., 1996). According to

(Abdeljaouad et al., 2020), to calculate the LB under uncertainty, we generate n scenario from the surgery durations and the recovery times, then calculate the LB for each scenario and the average value.

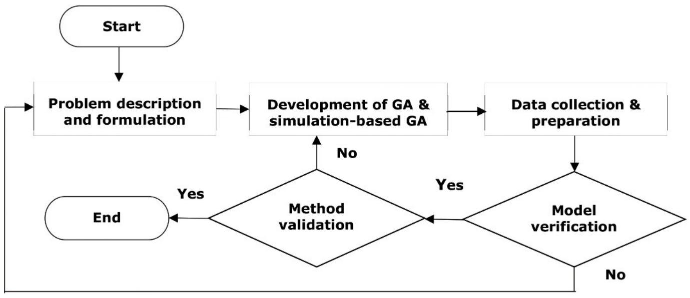

Figure 1. Flowchart of research methodology

2.1. Problem Description and Formulation

2.1.1. Problem Description

This study considers two-stage HFS scheduling problems for elective surgical cases: surgical procedure and recovery. For each day, J surgical cases must be allocated to ORs in the first stage and to PACU beds in the second stage and then scheduled. ORs are equipped with different equipment and have other characteristics. Surgery j can only be in the eligible OR meeting the required operation condition. The subset of operating room that are eligible to perform surgery j. According to (Vairaktarakis & Cai, 2003), the eligibility matrix A will be used to describe Mj, where $A[o_1,j] = 1$ if OR o1 is eligible to conduct surgery j otherwise $A[o_1, j]=0$. The propounded model aims to find a feasible scheduling that minimizes makespan. The following assumptions are made in the proposed model, as follows:

- Using an open scheduling policy that allows surgeons to share different ORs and move between them during the day.
- Only elective surgical cases were included without emergency cases.
- Human resources, materials, tools, and equipment are always available.
- Once started, any stage cannot be stopped or interrupted until it is finished.
- The time required to transfer the patient from the OR to recovery is neglected.
- All patients are ready for a surgical procedure on the scheduled day at the beginning.
- The priority is the same for all surgical cases, and cancellation is not allowed.

2.1.2. Parameters and Decision Variables Notations

Indices:

k Stage index ($k = 1$ is the first stage, and $k = 2$ is the second stage)

o_k OR or recovery bed index in stage k, $o_k = 1 \ldots, N_k$, where N_1 is the number of ORs and N_2 is the number of recovery beds in stage 2

j Surgery index $j = 1 \ldots, J$, where J is the number of surgical cases that needed schedule.

s Surgeon index $s = 1 \ldots, S$, where S is the number of surgeons

Parameters:

T_j^{set+cl} Clean time for OR after surgery j and set up time for next surgery.

T_s Setup time of surgeon s

T_j^k Duration time for surgery j in stage k

A Eligibility matrix of EM_j, where element $A[o_1, j] = 1$ if OR o1 is eligible for surgery j; 0 otherwise

M A big number

D_{js} Surgeon *s* is dedicated for surgery *j*

OT Opening Time for OR

Decision variable:

C_{max} Makespan

x_{ijso_1} 1 if surgery *i* precedes surgery *j* on OR *o* in stage 1, and it is implemented by surgeon *s*

0 Otherwise

x_{ijo_2} 1 if surgery *i* precedes surgery *j* on recovery bed *o* in stage 2

0 Otherwise

$T_j^{st/k}$ Starting time for surgery *j* in stage *k*

y_{jso_1} 1 if surgery *j* is assigned to OR *o* in stage 1

0 Otherwise

2.1.3. Mathematical Formulation

The objective function of the MILP minimizes the makespan, as given in Equation 1. Equation 2 calculates the completion times for the surgeries assigned to the recovery beds in stage 2. Equation 3 determines the starting times for the surgeries in stage 2. Equation 4 guarantees that the starting times for the surgeries in stage 1 don't violate the opening times. Equation 5 ensures surgery *i* precedes surgery *j*, where each surgical case can only be assigned to one OR in stage 1 and implemented by one surgeon, while it is assigned to one recovery bed in stage 2. Equation 6 ensures that each surgical case can only be assigned to one OR and one recovery bed in the PACU and implemented by one surgeon. Equation 7 ensures the eligibility of the ORs to perform the surgical cases assigned to them in stage 1. Equations 8 and 9 guarantee that the OR in stage 1 has one surgeon at a time and ensure that one surgery is performed from the available surgeries (disjunctive constraints). Equations 10 and 11 ensure that each recovery bed in stage 2 has one surgery at a time and the overlapping among surgeries on the same recovery bed is not allowed (disjunctive constraints). Equation 12 avoids the overlap among surgeries assigned to the same surgeon in stage 1, considering the setup time of the surgeon between successive surgeries. Equation 13 guarantees that each surgeon can implement their assigned surgery. Equations 14 and 15 are binary decision variables.

Objective Function:

$$minimize\ C_{max} \tag{1}$$

Subject to:

$$C_{max} \geq T_j^{st/k} + T_j^k \qquad j = 1 \ldots J \qquad k\ is\ the\ last\ stage \tag{2}$$

$$T_j^{st/2} \geq T_j^{st/1} + T_j^1 \qquad j = 1 \ldots J \tag{3}$$

$$T_j^{st/1} \geq OT \qquad j = 1 \ldots J \tag{4}$$

$$\begin{cases} \sum_{s=1}^{S} \sum_{o_1}^{N_1} x_{ijso_1} \leq 1 \\ \sum_{o_2}^{N_2} x_{ijo_2} \leq 1 \end{cases} \qquad i \neq j \qquad i = 1 \ldots J \qquad j = 1 \ldots J \tag{5}$$

$$x_{ijso_k} \leq y_{jso_k}, \qquad k = 1 \quad i = 1 \ldots J \quad j = 1 \ldots J \quad s = 1 \ldots S \tag{6}$$

$$y_{jso_1} \leq A[o_1, j] \qquad j = 1 \ldots J \quad s = 1 \ldots S \tag{7}$$

$$T_j^{st/1} + M \times (1 - x_{ijso_1}) \geq T_i^{st/1} + T_i^1 + T_i^{set+cl} \quad \begin{array}{c} i \neq j \quad i = 1 \dots J \\ j = 1 \dots J \quad s = 1 \dots S \\ i \& j \ disjunctive\ surgeries \end{array} \tag{8}$$

$$T_i^{st/1} + M \times x_{ijso_1} \geq T_j^{st/1} + T_j^1 + T_j^{set+cl} \quad \begin{array}{c} i \neq j \quad i = 1 \dots J \\ j = 1 \dots J \quad s = 1 \dots S \\ i \& j \ disjunctive\ surgeries \end{array} \tag{9}$$

$$T_j^{st/2} + M \times (1 - x_{ijo_2}) \geq T_i^2 + T_i^{st/2} \quad \begin{array}{c} i \neq j, i = 1 \dots J \\ j = 1 \dots J \\ i \& j \ disjunctive\ surgeries \end{array} \tag{10}$$

$$T_i^{st/2} + M \times x_{ijo_2} \geq T_j^2 + T_j^{st/2} \quad \begin{array}{c} i \neq j, i = 1 \dots J \\ j = 1 \dots J \\ i \& j \ disjunctive\ surgeries \end{array} \tag{11}$$

$$T_j^{st/1} + M \times x_{ijso_1} \geq T_i^{st/1} + T_i^1 + T_s \times y_{iso_1} \quad \begin{array}{c} i = 1 \dots J \\ j = 1 \dots J \quad s = 1 \dots S \end{array} \tag{12}$$

$$y_{jso_1} \leq D_{js} \quad j = 1 \dots J \quad s = 1 \dots S \tag{13}$$

$$x_{ijso_k} \in \{0,1\} \quad \begin{array}{c} i \neq j \quad i = 1 \dots J \\ j = 1 \dots J \quad k = 1, 2 \\ s = 1 \dots S \end{array} \tag{14}$$

$$y_{jso_k} \in \{0,1\} \quad j = 1 \dots J \quad k = 1, 2 \quad s = 1 \dots S \tag{15}$$

2.2. Simulation-Based Optimization

The framework of a simulation-based GA is given in Figure 2, while Table 2 shows the pseudo-code. GA is one of the optimization and search algorithms introduced by John Holland in the 1970s (Kumar et al., 2021). It is derived from Darwin's evolutionary theory of survival of the fittest and biological evolution (Katoch et al., 2020). GA involves abstracting the problem space as a set of individuals and then exploring the fittest individual by iteratively producing generations. The integration of Monte Carlo simulation with the GA serves not merely to reproduce deterministic outcomes using random inputs but to evaluate the robustness of each candidate schedule under stochastic conditions. Because the makespan objective is a nonlinear function of surgery durations and recovery times constrained by no-wait and eligibility rules, the expected makespan cannot be obtained by substituting the mean parameter values into the deterministic model (i.e., $E[f(X)] \neq f(E[X])$. The simulation-based approach, therefore, captures the propagation and interaction of variability across both surgery and recovery stages, which directly influence resource blocking and stage synchronization. This enables identification of solutions that remain effective under variability, not just under average conditions. In addition, the simulation generates confidence intervals and stability metrics that are otherwise analytically intractable, thereby enhancing the practical interpretability of the proposed scheduling framework.

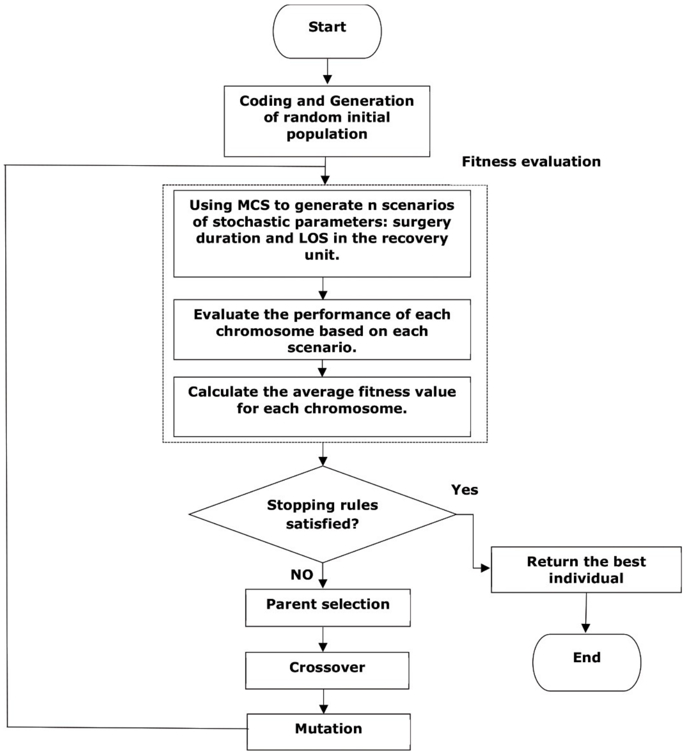

Figure 2. Framework of simulation-based GA

2.2.1. Coding and Generation of Initial Population

Randomly generate an initial population, which contains P_{size} chromosomes. Each represents a feasible solution for the problem and is coded as an integer array like the method used in (Fei et al., 2010). It has four parts as shown in Figure 3, part 1 is a vector of size equal to the number of patients (J), it records the order of patients passing through the ORs. Part 2 is a vector of size J. The numbers refer to the recovery bed of each patient in the same order in the first part. Part 3 is a vector of size equal to OR number (N_1)-1. The numbers refer to the delimitation positions that show which patients in part 1 are assigned to each OR. Part 4 is a vector of size J; it shows the order in the patient passing through the recovery beds.

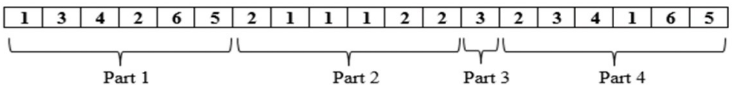

Figure 3. Coding scheme for two stages-scheduling problem.

2.2.2. Fitness Evaluation and Simulation

After generating the initial population, the performance of each chromosome is evaluated by calculating its fitness value, which is defined as the makespan. The MCS model is used to generate n scenarios of stochastic parameters and then evaluate the performance of each individual in the population based on each scenario. Each chromosome is decoded into a complete timeline by defining the patient's sequence, their assigned operating rooms, and recovery beds.

To handle constraint violations, a combination of rejection and repair strategies is adopted during the decoding and schedule construction phase. The operating room eligibility constraint is treated as a strict constraint; if a surgery is assigned to a non-eligible operating room, the chromosome is considered infeasible and is directly rejected. For the remaining constraints, including operating room availability, surgeon availability, recovery bed availability, and no-wait constraints, a repair mechanism is applied. The complete chromosome decoding and constraint-repair procedure, together with a step-by-step worked example illustrating the resolution of surgeon-availability and no-wait conflicts, is provided in Appendix A.

The timeline is then simulated chronologically, with surgeries performed according to the assigned operating room sequence. Upon completion, patients are immediately transferred to the recovery room, subject to a no-waiting constraint. If no recovery bed is available at the expected completion time of a surgery, the start of the procedure is delayed until a bed becomes available immediately afterward. This procedure ensures that all resource and priority constraints are met. The total completion time is calculated as the time it takes for the last patient to leave the recovery room. Finally, the chromosome fitness value is calculated as the average of the total completion time across all simulated scenarios.

Since fitness is estimated through simulation, the number of Monte Carlo replications (n) affects the accuracy of the estimated makespan. Increasing n improves the precision of the fitness estimate but increases computational time. Therefore, an adaptive procedure was adopted. The simulation evaluation starts with an initial sample of 30 replications, and the relative error is calculated as the half-width of the 95% confidence interval divided by the sample mean. Additional replications are generated in increments until the relative error falls below 5%. This criterion provides a practical trade-off between computational effort and statistical precision, ensuring sufficient accuracy while minimizing computational overhead.

The stochastic parameters in the simulation model, namely surgery duration and length of stay (LOS) in the recovery unit, were derived from real hospital data. A data fitting procedure was conducted to identify appropriate probability distributions for each surgical specialty. Specifically, the Input Analyzer tool in Arena software was used to fit candidate probability distributions to the empirical data. The selection of the best-fitting distribution was based on the Kolmogorov–Smirnov (K–S) goodness-of-fit test. For each fitted distribution, the null hypothesis that the data follows the specified distribution was tested at a significance level of 0.05. Since all obtained p-values were greater than 0.05, the null hypothesis could not be rejected, indicating that the selected distributions provide an adequate fit to the observed data.

Different distributions were identified by surgical specialty, including Normal, Lognormal, Beta, Gamma, and Triangular distributions. This approach allows the simulation model to capture the variability and heterogeneity in surgery durations and recovery times across different types of procedures, thereby improving the realism and accuracy of the simulation outcomes. The fitted distributions were then used within the Monte Carlo simulation to generate random scenarios for each chromosome evaluation in the genetic algorithm.

2.2.3. Parent Selection

The roulette wheel selection method is used to select two parents from the population for the next generation. In this method, the chance of survival for the chromosome with a better fitness value is higher than that of the weaker ones (Fei et al., 2010).

2.2.4. Crossovers

A new child with probability (P_c) is formed in this step by combining the two selected parents. Two different crossover operators stochastically were used. Either for sequencing patients in ORs (part 1) will apply the OX crossover operator (Poon & Carter, 1995), or for patients' assignment to recovery beds (part 2) will apply the two-point crossover operator (Souki et al., 2009). Since the crossover acts on one part of the chromosome, other parts should be modified to adhere to all constraints related to our problem.

2.2.5. Mutation

This step includes a random selection of one of the children resulting from crossover. Then, the selected children are mutated with a probability P_m to generate a new chromosome. Three different mutation operators will be randomly applied on different parts of the chromosome: A mutation operator for part 1 includes selecting two elements randomly in the first part and switching between them. A mutation operator for part 2 includes selecting two elements with different values in the second part and switching between them. A mutation operator for part 3, which includes randomly generating an integer $f \in \{1\ldots, (N1\text{-}1)\}$, then eliminates f_{th} element from part 3 so that the size of the vector becomes equal (N1-2). After that, generate a random integer number $k \in \{1..., J\}$ and insert it into part 3, then arrange the elements in this part in ascending order.

2.2.6. Termination Criteria

The stopping algorithm criterion adopted in this study is reaching the maximum number of iterations without improving the solutions (Lin & Yen, 2023). If the algorithm runs at the maximum number of the previously specified iteration without improvement, the algorithm will stop and return the best solution, defined as the chromosome with the minimum average makespan over all Monte Carlo simulation scenarios.

Simulation-based genetic algorithm

```
Initialize Psize, PC, Pm, n, MAX_ITERATION
Generate a random initial population, which includes Psize chromosomes
For each chromosome
    Generate n scenarios of surgery durations and LOS in PACU
    Fitness evaluation using each scenario
    Calculate average for fitness values
End for
current_iteration= 0
Pervious_solution=100
While current_iteration < MAX_ITERATION
      Child= selection of two parents using the roulette wheel method
      C= random generated number (0,1)
      if C < 1- PC
        Randomly perform OX crossover or two-point crossover between parents
        Update child
      End if
      M= random generated number (0,1)
      if M<= Pm
        Perform mutation on one of the children
        Update child
      End if
      Check on the feasibility for each child
       for each feasible child
          Generate n scenarios of surgery durations and LOS in PACU
          Fitness evaluation using each scenario
         Calculate average for fitness values
     End for
     Update population # To include the best chromosomes that have the lowest makespan from the population and child
     Best_solution = minimum fitness value for the population
     If Best_solution = Pervious_solution
       current_iteration = current_iteration+1
     Else
       current_iteration = 0
     End if
End while
```

Table 2. Pseudo code for simulation-based GA

3. Data Collection

In a general hospital in Kuwait, there are twelve surgical specialties. The hospital's surgical suite comprises five ORs (heterogeneous) and four recovery beds. The ORs open simultaneously with the recovery room for ten hours, from 8 am to 8 pm. It remains open until the last patient is transferred from the intensive care unit. Overtime may be required to complete all surgical cases scheduled for the day. The dataset collected includes 1259 historical records obtained from the hospital database. This data contains the type of surgery for each case, the date, the name of the surgeon, and the start and end times of the surgery and recovery. After each surgical case, half an hour is allocated to clean the OR and set it for the next surgery. In addition, half an hour is allocated for the surgeon after completing the operation to rest and prepare for subsequent surgeries.

4. Results and Discussion

The CPLEX solver and Python codes were run by a computer with Windows 10 pro with 6 GB of memory and Intel Core i5 @ 2.40 GHz. Final selection of parameters, initial population (P_{size})= 50 for small cases and 100 for real cases, mutation rate (P_m)= 0.1, crossover rate (P_c)= 0.8, and maximum iteration 200 for small cases and 1000 for real cases.

4.1. Simulation Model Verification and Validation

Before integrating the simulation model with the genetic algorithm, the model was independently verified and validated. Model verification was conducted by examining the simulation logic and generated schedules to ensure compliance with all problem constraints, including operating room eligibility, surgeon availability, recovery bed availability, and the no-wait requirement between surgery and recovery stages. Several test instances were manually checked to confirm that the simulation outputs were consistent with the scheduling rules and mathematical assumptions.

Model validation was performed by comparing simulated makespan estimates against actual hospital scheduling outcomes using the confidence interval validation procedure recommended by Banks et al. (2005). This approach evaluates model accuracy through a two-step decision framework. First, the actual hospital makespan μ_0 must fall within the simulated model's 95% confidence interval. Second, the worst-case error, which is the distance from μ_0 to the farthest confidence interval endpoint, must be within an acceptable tolerance ε. This criterion is more rigorous than simply checking whether μ_0 falls within the confidence interval, as it additionally ensures that the model's maximum possible error remains below the tolerance threshold. By satisfying both conditions, the model is deemed sufficiently accurate to serve as the valid evaluation engine for the simulation-based genetic algorithm.

All five validation cases satisfy the acceptance criterion. As shown in Table 3, each actual hospital makespan falls within the 95% confidence interval of the simulation, and the worst-case error remains within the acceptable tolerance of 1 hour. These results confirm that the simulation model accurately reproduces the behavior of the real system under uncertainty. The model's predictions remain within operationally acceptable bounds, establishing its validity for use as the evaluation engine within the simulation-based genetic algorithm.

Real instance	Mean makespan (hr)	Sample size (n)	Margin of error (hr)	Relative error (%)	CI lower bound (hr)	CI upper bound (hr)	Actual μ_0 (hr)	Banks verdict
1	13.382	30	0.506	3.779	12.877	13.888	13.860	Accept
2	14.576	30	0.413	2.832	14.163	14.989	14.692	Accept
3	13.105	30	0.371	2.830	12.734	13.476	12.752	Accept
4	13.380	50	0.578	4.322	12.802	13.958	13.007	Accept
5	13.186	30	0.450	3.412	12.736	13.636	13.505	Accept

Table 3. Validation Results: Simulated Makespan vs. Actual Hospital Makespan

4.2. Small Size Instances

Seven randomly generated small instances with different resource sizes are utilized to evaluate the propounded MILP and the proposed methods (GA and simulation-based GA), as shown in Table 4. The results of CPLEX and GA for the seven instances under a deterministic environment (the mean values) for each surgical specialty are summarized in Table 5.

Instance no	Number of			
	Operating rooms	Recovery beds	Surgeons	Surgical cases
1	2	2	3	6
2	2	1	3	6
3	2	2	4	8
4	2	1	4	8
5	3	3	5	10
6	3	2	5	10
7	3	3	8	12

Table 4. Test instances information

Instance no	LB (hr)	MILP			GA		
		Makespan (hr)	Relative gap (%)	CPU time (second)	Makespan (hr)	Relative gap (%)	CPU time (second)
1	8.580	9.060	5.590	0.230	9.060	5.590	3
2	9.880	10.240	3.640	0.750	10.240	3.640	4
3	10.870	10.880	0.092	1.540	10.880	0.092	4
4	12.560	12.780	1.752	2.240	12.890	2.627	5
5	8.833	9.060	2.570	30.300	9.400	6.419	21
6	8.480	9.380	10.610	440.930	9.380	10.610	28
7	9.787	9.850	0.644	1717.160	10.090	3.096	66
Average	**9.856**	**10.179**	**3.557**	**313.307**	**10.277**	**4.582**	**18.714**

Table 5. Results of small instances with deterministic (Average value).

The results of CPLEX illustrated that the CPU time to obtain the optimal solution increases as the number of surgeons and surgical cases increases. Also, the complexity of the problem increases when the number of recovery beds is less than the number of ORs. Therefore, for the seven instances, the optimal solution can be obtained in a reasonable time. According to (Abdeljaouad et al., 2020), the time is considered reasonable if it is half an hour or less. Nevertheless, cases that take more than half an hour, such as medium or large-scale instances, can be dealt with using heuristics or metaheuristics that can reach a satisfactory solution quickly.

However, the obtained results led to several observations. Firstly, in some instances, the GA could duplicate the results of the mathematical model. Secondly, the average relative gap between the LB and optimal solution is 3.557 %, while it is 4.582% between the LB and GA solutions. The relative gap between a given solution and the lower bound was calculated as the difference between the solution value and the lower bound, divided by the lower bound, and then multiplied by 100 to express the result as a percentage. Thirdly, the CPU time for the GA is low, ranging between 6 to 66 seconds. Meanwhile, the CPU time increased significantly in the MILP, CPLEX, as the complexity of the problem increased. Thus, it is clear that the GA can obtain high-quality solutions efficiently for small instances in a reasonable time.

Note that all relative gap values reported in the following tables are calculated for each instance using full-precision data prior to rounding. The reported average gap represents the mean of instance-level gaps; therefore, it may not exactly match the gap computed from the average lower bound and average makespan values.

To ensure statistical significance, the relative error was computed as the half-width of the 95% confidence interval divide d by the sample mean. As reported in Table 6, the relative error remained below 5% across all instances, confirming that the selected number of replications yields sufficient precision: the estimated makespan lies within ±5% of the true mean with 95% confidence.

The simulation-based GA results for seven small instances under uncertainty are summarized in Table 6. On average, the obtained solutions deviate from the lower bound by 6.531%, with a mean CPU time of 56.286 seconds, demonstrating that the algorithm produces near-optimal solutions within a reasonable computational budget.

Instance no	LB (hr)	Makespan (hr)	Relative error (%)	Relative gap (%)	CPU time (second)
1	8.589	8.874	3.765	3.318	25
2	9.731	10.480	3.287	7.697	27
3	10.457	10.714	4.820	2.458	41
4	12.303	13.043	3.082	6.015	77
5	8.202	8.993	3.845	9.644	44
6	8.701	9.544	3.397	9.689	93
7	9.422	10.072	2.903	6.899	87
Average	**9.629**	**10.246**	**3.586**	**6.531**	**56.286**

Table 6. Results of small instances with uncertainty

4.3. Real Instances

Five real hospital instances were solved under certainty and uncertainty, as shown in Tables 7 and 8. These results were compared with the actual schedule prepared by the hospital administrators. Based on the GA results shown in Table 8, the average makespan is 9.530 hours. If the first surgery begins at 8 am, the last surgery will end before 8 pm to avoid exceeding the official working hours. Also, the average relative gap between makespan and the LB is 4.053 %, which indicates a relatively small deviation. This indicates that the GA could acquire the optimal results in a reasonable time, an average of 71 seconds.

Table 9 represents the results obtained using simulation-based GA; the average range makespan of 10.779 hours necessitates an average of 0.779 hours of overtime. The average relative gap is 13.735%, and 4.053%, respectively. In this regard, the relative gap increases when simulation-based GA is utilized to solve the surgical scheduling problem. This escalation could be attributed to the significant influence of uncertainty, such as unexpected events like medical complications, which can extend surgery durations and disrupt scheduling, amplifying the relative gap. In conclusion, the simulation-based GA solution approach has shown promise in effectively managing uncertainty in hospital scheduling.

The obtained results also illustrated that the average CPU time is 264.4 seconds (4.407 minutes). Additionally, the average makespan of the actual scheduling is 13.563 hours; thus, the overtime is high, where the working time is 10 hours per day. Meanwhile, the average makespan for the simulation-based GA is 10.779 hours. The average relative gap between the actual scheduling and LB equals 43.284%, which is relatively large. The paired t-test is implemented to check whether the difference between the makespan values is significant; the result showed that the p-value is 0.001 less than the significant level (α =0.05). Hence, we reject the null hypothesis, and there is a significant difference between makespan for the simulation-based GA and makespan for the actual scheduling.

Therefore, the simulation-based GA demonstrates better performance than the actual scheduling prepared by the hospital team, with makespan reductions ranging between 11.771% and 25.080%.

Real Instance no	Number of			
	Operating rooms	Recovery beds	Surgeons	Surgical cases
1	5	4	13	20
2	5	4	12	20
3	5	4	11	20
4	5	4	12	17
5	5	4	11	20

Table 7. Real instances information

Real Instance no	LB (hr)	GA		
		Makespan (hr)	Relative gap (%)	CPU time (second)
1	9.116	9.460	3.774	74
2	10.028	10.390	3.610	71
3	9.386	9.900	5.480	74
4	8.654	8.850	2.265	64
5	8.608	9.050	5.135	72
Average	**9.158**	**9.530**	**4.053**	**71**

Table 8. Results of real instances with deterministic

Real instance no	LB	Simulation-based GA				Actual scheduling		Reduction (%)
		Makespan (hr)	Relative error (%)	Relative gap (%)	CPU time (second)	Makespan (hr)	Relative gap (%)	
1	9.792	11.108	2.511	13.440	227	13.860	41.544	19.856
2	10.248	11.299	2.688	10.256	320	14.692	43.365	23.094
3	9.677	11.251	2.899	16.265	189	12.752	31.776	11.771
4	9.037	10.120	3.182	11.984	243	13.007	43.931	22.196
5	8.668	10.118	2.316	16.728	342	13.505	55.803	25.080
Average	**9.484**	**10.779**	**2.719**	**13.735**	**264.200**	**13.563**	**43.284**	**20.399**

Table 9. Results of real instances under uncertainty

5. Conclusions

This study addressed the two-stage elective surgery scheduling problem considering no-wait constraints between surgery and recovery. A mixed-integer linear programming (MILP) model was formulated to minimize makespan while integrating key practical constraints such as operating room (OR) eligibility, recovery bed availability, and surgeon dedication. To solve the problem efficiently, a Genetic Algorithm (GA) was developed for deterministic conditions, and a simulation-based GA was proposed to handle uncertainty in surgery duration and recovery time. The results demonstrated that the GA consistently provided optimal or near-optimal solutions for small instances, achieving an average relative gap of 4.053% and an average computational time of 71 seconds for real cases. Under uncertainty, the simulation-based GA achieved an average relative gap of 13.735%, mainly due to the propagation of stochastic variability, yet still produced substantially improved schedules compared with actual hospital operations. Specifically, the simulation-based GA reduced makespan by 11.77%–25.08%, highlighting its effectiveness in improving hospital resource utilization and reducing overtime.These findings validate that the integration of simulation with GA enhances robustness against variability in surgery and recovery durations, allowing administrators to generate schedules that remain feasible and efficient under uncertainty. The study thus bridges the gap between deterministic optimization and real-world stochastic scheduling in healthcare environments.From a methodological perspective, the proposed framework contributes by combining a two-stage no-wait hybrid flow shop model with stochastic simulation, an approach that can be generalized to other healthcare or manufacturing systems with interdependent stages and uncertain processing times.Future research could extend this work by incorporating the pre-operative stage to form a complete three-stage system, exploring larger-scale instances, and evaluating alternative metaheuristics such as Particle Swarm Optimization, Red Deer Algorithm, or hybrid deep-learning-assisted approaches to further enhance computational efficiency and predictive accuracy

Declaration of Conflicting Interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

Authors' contributions

Lawrence Al-Fandi: Conceptualization, Methodology, Formal analysis, Investigation, Project administration, Supervision, Writing - review & editing.

Rahaf Abedalrohman: Conceptualization, Methodology, Formal analysis, Investigation, Visualization, Writing - original draft, Writing - review & editing.

Najat Almasarwah: Methodology, Writing - review & editing.

Data availability

Data available upon request

Use of Artificial Intelligence

The authors used ChatGPT and Grammarly for language editing, including grammar and readability improvements. All scientific content was written, verified, and approved by the authors, who take full responsibility for the manuscript.

References

Abdeljaouad, M. A., Bahroun, Z., Saadani, N. E. H., & Zouari, B. (2020). A simulated annealing for a daily operating room scheduling problem under constraints of uncertainty and setup. *INFOR Information Systems and Operational Research, 58(*3), 456–477. https://doi.org/10.1080/03155986.2020.1734901

Abdeljaoued, M. A., Bahroun, Z., & Saadani, N. E. H. (2021). Short-term operating room scheduling: a parallel machine under resource constraints problem. *International Journal of Operational Research, 40*(2), 162–184. https://doi.org/10.1504/IJOR.2021.113501

Aringhieri, R., Landa, P., Soriano, P., Tanfani, E., & Testi, A. (2015). A two level metaheuristic for the operating room scheduling and assignment problem. *Computers & Operations Research, 54,* 21–34. https://doi.org/10.1016/j.cor.2014.08.014

Azaiez, M. N., Gharbi, A., Kacem, I., Makhlouf, Y., & Masmoudi, M. (2021). Two-stage no-wait hybrid flow shop scheduling with inter-stage flexibility: A mathematical model. In *2021 IEEE International Conference on Networking, Sensing and Control (ICNSC),* (pp. 1–5). Xiamen, China. https://doi.org/10.1109/icnsc52481.2021.9702162

Bai, M., Storer, R. H., & Tonkay, G. L. (2017). A sample gradient-based algorithm for a multiple-OR and PACU surgery scheduling problem. *IISE Transactions, 49*(4), 367–380. https://doi.org/10.1080/0740817x.2016.1237061

Bam, M., Denton, B. T., Van Oyen, M. P., & Cowen, M. E. (2017). Surgery scheduling with recovery resources. *IISE Transactions*, *49*(10), 942–955. https://doi.org/10.1080/24725854.2017.1325027

Banks, J., Carson, J. S., Nelson, B. L., & Nicol, D. M. (2005). *Discrete-event system simulation* (4th ed.). Pearson Education Inc.

Bektur, G., & Aslan, H. K. (2024). Artificial bee colony algorithm for operating room scheduling problem with dedicated/flexible resources and cooperative operations. *An International Journal of Optimization and Control, 14*(3), 193–207. https://doi.org/10.11121/ijocta.1466

Clavel, D., Mahulea, C., Albareda, J., & Silva, M. (2020). A Decision Support System for Elective Surgery Scheduling under Uncertain Durations. *Applied Sciences*, *10*(6), 1937. https://doi.org/10.3390/app10061937

Denton, B., Viapiano, J., & Vogl, A. (2007). Optimization of surgery sequencing and scheduling decisions under uncertainty. *Health Care Management Science*, *10*(1), 13–24. https://doi.org/10.1007/s10729-006-9005-4

Díaz-López, D. M., López-Valencia, N. A., González-Neira, E. M., Barrera, D., Suárez, D. R., Caro-Gutiérrez, M. P., & Sefair, C. (2018). A simulation-optimization approach for the surgery scheduling problem: a case study considering stochastic surgical times. *International Journal of Industrial Engineering Computations, 9,* 409–422. https://doi.org/10.5267/j.ijiec.2018.1.002

Dolatkhah, M., Hashemi-Doulabi, H., Rei, W., & Gendreau, M. (2026). A reinforcement-learning-based column generation algorithm for integrated operating room planning and scheduling. *International Journal of Production Research,* 1-41. https://doi.org/10.1080/00207543.2026.2637778

Farzad, G., & Mohammad, S. M. (2016). A stochastic surgery sequencing model considering the moral and human virtues. *Modern Applied Science*, *10*(9), 68. https://doi.org/10.5539/mas.v10n9p68

Fei, H., Meskens, N., & Chu, C. (2010). A planning and scheduling problem for an operating theatre using an open scheduling strategy. *Computers & Industrial Engineering*, *58*(2), 221–230. https://doi.org/10.1016/j.cie.2009.02.012

Guinet, A., Solomon, M., Kedia, P., & Dussauchoy, A. (1996). A computational study of heuristics for two-stage flexible flowshops. *International Journal of Production Research*, *34*(5), 1399–1415. https://doi.org/10.1080/00207549608904972

Gul, S., Denton, B. T., Fowler, J. W., & Huschka, T. (2011). Bi□Criteria scheduling of surgical services for an outpatient procedure center. *Production and Operations Management*, *20*(3), 406–417. https://doi.org/10.1111/j.1937-5956.2011.01232.x

Guzman, E., Andres, B., & Poler, R. (2022). Models and algorithms for production planning, scheduling and sequencing problems: A holistic framework and a systematic review. *Journal of Industrial Information Integration,* 27, 100287.

Hsu, P., Lo, S., Hwang, H., & Lin, B. M. T. (2021). Scheduling of anaesthesia operations in operating rooms. *Healthcare*, *9*(6), 640. https://doi.org/10.3390/healthcare9060640

Huang, G. X., Xiang, W., Li, C., Zheng, Q., Zhou, S., Shen, B. Q., & Chen, S. F. (2012). Surgical scheduling based on hybrid Flow-Shop scheduling. *Applied Mechanics and Materials*, *201–202*, 1004–1007. https://doi.org/10.4028/www.scientific.net/amm.201-202.1004

Kamran, M. A., Karimi, B., & Dellaert, N. (2018). Uncertainty in advance scheduling problem in operating room planning. *Computers & Industrial Engineering*, *126*, 252–268. https://doi.org/10.1016/j.cie.2018.09.030

Katoch, S., Chauhan, S. S., & Kumar, V. (2020). A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications, 80*(5). https://doi.org/10.1007/s11042-020-10139-6

Khaniyev, T., Kayış, E., & Güllü, R. (2020). Next-day operating room scheduling with uncertain surgery durations: Exact analysis and heuristics. *European Journal of Operational Research*, *286*(1), 49–62. https://doi.org/10.1016/j.ejor.2020.03.002

Kumar, R., Memoria, M., Gupta, A., & Awasthi, M. (2021). Critical Analysis of Genetic Algorithm under Crossover and Mutation Rate. In *2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)* (pp. 976–980). https://doi.org/10.1109/icac3n53548.2021.972564

Landa, P., Aringhieri, R., Soriano, P., Tànfani, E., & Testi, A. (2016). A hybrid optimization algorithm for surgeries scheduling. *Operations Research for Health Care*, *8*, 103–114. https://doi.org/10.1016/j.orhc.2016.01.001

Lin, Y. K., & Chong, C. S. (2025). Solving Three-Stage Operating Room Scheduling Problems with Uncertain Surgery Durations. *Mathematics, 13*(12), 1973. https://doi.org/10.3390/math13121973

Lin, Y., & Chou, Y. (2019). A hybrid genetic algorithm for operating room scheduling. *Health Care Management Science*, *23*(2), 249–263. https://doi.org/10.1007/s10729-019-09481-5

Lin, Y., & Yen, C. (2023). Genetic algorithm for solving the No-Wait Three-Stage surgery scheduling problem. *Healthcare*, *11*(5), 739. https://doi.org/10.3390/healthcare11050739

Park, J., Kim, B., Eom, M., & Choi, B. K. (2021). Operating room scheduling considering surgeons' preferences and cooperative operations. *Computers & Industrial Engineering*, *157*, 107306. https://doi.org/10.1016/j.cie.2021.107306

Patrão, R. L., Garcia, R. C., & Da Silva, J. M. (2022). An Integrated Two-Level Integer Linear Program (ILP) model for elective surgery scheduling: a case study in an Italian hospital. *Mathematics*, *10*(11), 1901. https://doi.org/10.3390/math10111901

Poon, P., & Carter, J. (1995). Genetic algorithm crossover operators for ordering applications. *Computers & Operations Research*, *22*(1), 135–147. https://doi.org/10.1016/0305-0548(93)e0024-n

Rahimi, I., & Gandomi, A.H. (2021). A comprehensive review and analysis of Operating Room and Surgery Scheduling. *Archives of Computational Methods in Engineering, 28,* 1667–1688. https://doi.org/10.1007/s11831-020-09432-2

Ramadan, S., Abu-Shams, M., Al-Dahidi, S., Odeh, I., & Almasarwah, N. (2025). A data-driven approach for predicting remaining intra-surgical time and enhancing operating room efficiency. *Journal of Industrial Engineering and Management,* 18(1), 145-166. https://doi.org/10.3926/jiem.8543

Saremi, A., Jula, P., ElMekkawy, T., & Wang, G. G. (2013). Appointment scheduling of outpatient surgical services in a multistage operating room department. *International Journal of Production Economics*, *141*(2), 646–658. https://doi.org/10.1016/j.ijpe.2012.10.004

Shehadeh, K. S., & Padman, R. (2022). Stochastic optimization approaches for elective surgery scheduling with downstream capacity constraints: Models, challenges, and opportunities. *Computers & Operations Research*, *137*, 105523. https://doi.org/10.1016/j.cor.2021.105523

Souki, M., Youssef, S. B., & Rebai, A. (2009). Memetic algorithm for operating room admissions. In *2009 International Conference on Computers & Industrial Engineering* (pp. 519–524). IEEE. https://doi.org/10.1109/iccie.2009.5223833

Subramanian, D., Pekny, J. F., & Reklaitis, G. V. (2001). A simulation□ optimization framework for research and development pipeline management. *AIChE Journal*, *47*(10), 2226–2242. https://doi.org/10.1002/aic.690471010

Swilugar, A., & Herliansyah, M. K. (2025). Operating Room Scheduling Optimization Under Surgeon and Nurse Constraints Using Genetic Algorithm. *TIERS Information Technology Journal, 6*(2), 158–170. https://doi.org/10.38043/tiers.v6i2.7164

Tayyab, A., & Saif, U. (2022). A Two-Stage genetic artificial bee colony algorithm for solving integrated operating room planning and scheduling problem with capacity constraints of downstream wards. *IEEE Access*, *10*, 131109 –131127. https://doi.org/10.1109/access.2022.3229709

Tayyab, A., Ullah, S., & Baki, M. F. (2023). An Outer Approximation Method for Scheduling Elective Surgeries with Sequence Dependent Setup Times to Multiple Operating Rooms. *Mathematics*, *11*(11), 2441. https://doi.org/10.3390/math11112441

Turhan, A. M., & Bilgen, B. (2022). A mat-heuristic based solution approach for an extended nurse rostering problem with skills and units. *Socio-Economic Planning Sciences, 82,* 101300.

Vairaktarakis, G. L., & Cai, X. (2003). The value of processing flexibility in multipurpose machines. *IIE Transactions*, *35*(8), 763–774. https://doi.org/10.1080/07408170304349

Varmazyar, M., Akhavan-Tabatabaei, R., Salmasi, N., & Modarres, M. (2020). Operating room scheduling problem under uncertainty: Application of continuous phase-type distributions. *IISE Transactions*, *52*(2), 216–235. https://doi.org/10.1080/24725854.2019.1628372

Vieira, B., Demirtas, D., van de Kamer, J. B., Hans, E. W., Jongste, W., & van Harten, W. (2021). Radiotherapy treatment scheduling: Implementing operations research into clinical practice. *Plos One, 16*(2), e0247428.

Wang, K., Qin, H., Huang, Y., Luo, M., & Zhou, L. (2020). Surgery scheduling in outpatient procedure centre with re-entrant patient flow and fuzzy service times. *Omega*, *102*, 102350. https://doi.org/10.1016/j.omega.2020.102350

Wang, S., Su, H., & Wan, G. (2015). Resource-constrained machine scheduling with machine eligibility restriction and its applications to surgical operations scheduling. *Journal of Combinatorial Optimization*, *30*(4), 982–995. https://doi.org/10.1007/s10878-015-9860-3

Wang, S., Su, H., Wan, G., & Zhong, L. (2022). Surgery scheduling in the presence of operating room eligibility and dedicated surgeon: an adaptive composite dispatching method. *International Journal of Production Research*, *61*(6), 1866–1881. https://doi.org/10.1080/00207543.2022.2050827

Wang, Y., Tang, J., Pan, Z., & Yan, C. (2014). Particle swarm optimization-based planning and scheduling for a laminar-flow operating room with downstream resources. *Soft Computing*, *19*(10), 2913–2926. https://doi.org/10.1007/s00500-014-1453-z

Xiang, W., Yin, J., & Lim, G. (2015). An ant colony optimization approach for solving an operating room surgery scheduling problem. *Computers & Industrial Engineering, 85,* 335–345. https://doi.org/10.1016/j.cie.2015.04.0

Xiao, G., Van Jaarsveld, W., Dong, M., & Van De Klundert, J. (2016). Stochastic programming analysis and solutions to schedule overcrowded operating rooms in China. *Computers & Operations Research*, *74*, 78–91. https://doi.org/10.1016/j.cor.2016.04.017

Xue, J., Li, Z., & Zhang, S. (2025). Multi-resource constrained elective surgical scheduling with Nash equilibrium toward smart hospitals. *Scientific Reports, 15*(1), 3946. https://doi.org/10.1038/s41598-025-87867-y

Zhang, J., Dridi, M., & Moudni, A. E. (2020). Column-generation-based heuristic approaches to stochastic surgery scheduling with downstream capacity constraints. *International Journal of Production Economics*, *229*, 107764. https://doi.org/10.1016/j.ijpe.2020.107764