Journal of Industrial and Systems Engineering
Vol. 13, No. 4, pp. 262-287
Autumn (November) 2021

A new genetic algorithm to solve integrated operating room scheduling problem with multiple objective functions

Pardis Shahhosseini[1] , Mohammad Ali Beheshtinia[1*]

[1] *Industrial Engineering Department, Faculty of Engineering, Semnan University, Semnan, Iran*

pardis.shahhosseini@semnan.ac.ir, beheshtinia@semnan.ac.ir

Abstract

In this paper, a new genetic algorithm is presented to plan and schedule operating rooms at the operational level to minimize completion time, surgeons' free time window, and operating rooms' overtime, idle time, and setup time costs. The duration of surgeries is calculated according to a predetermined time plus an allowance related to the uncertainty of the surgery time. Also, the operating rooms' setup times depend on the sequence of surgeries. The time window constraint involves resource availability such as surgeons and operating rooms. First, a mixed-integer nonlinear mathematical model is proposed to solve the problem. Thereafter, a genetic algorithm is developed to solve the problem inspired from the role model concept in sociology using simulating and differentiating procedures, namely Role Model Genetic Algorithm (RMGA). The performance of the proposed algorithm is examined by comparing it with a conventional genetic algorithm and a hybrid genetic algorithm proposed for the nearest problem in the literature to the current problem. The results shows that RMGA prepares better results.

Keywords: Genetic algorithm, scheduling, operating room scheduling, multiple operating rooms

1-Introduction

The development of efficient health systems has become significant in recent decades due to the rapidly rising costs of healthcare in developed countries and the simultaneous growth in demand for healthcare and patient expectations of service quality. As a result, governments and health systems' decision-makers are constantly seeking to develop more efficient health systems (Ahmadi-Javid, Jalali, & Klassen, 2017; Hulshof, Kortbeek, Boucherie, Hans, & Bakker, 2012).Operating rooms are estimated to account for more than 40% of a hospital's total revenue and a significant portion of its total costs, making it the largest hospital cost center as well as the largest source of revenue. Therefore, its management and efficiency is the subject of a wide range of studies (Denton, Miller, Balasubramanian, & Huschka, 2010; Van Oostrum, Bredenhoff, & Hans, 2010; Zhu, Fan, Yang, Pei, & Pardalos, 2019). Despite the dramatic increase in demand for surgery in recent years, the capacity of operating rooms has not been improved in many cases(Conforti, Guerriero, & Guido, 2010). Recent studies show that over 60% of patients in each hospital require surgery. For most hospitals, good performance and high operating room efficiency play a key role in improving the benefits and quality of hospital services to the patient (Zhu et al., 2019). The planning and scheduling process, according to the definitions presented by Tan, El Mekkawy, Peng, and Oppenheimer (2007) is the adaptation of the supply and demand process, and sequencing and timing activities (Denton et al., 2010). Poor planning may lead to idle time intervals, excessive overwork, and delays or cancellations of surgeries, which are reflected in the additional costs and loss of revenue (Conforti et al., 2010; Latorre-Núñez et al., 2016) and are increasingly studied to provide more efficient solution methods.

*Corresponding author

In this study, a new genetic algorithm is proposed to solve a multi-objective operating room scheduling problem. The considered objective functions try to optimize the utilization of hospital resources and satisfy patient, and surgeon needs. The problem is addressed under the open schedule strategy, and the availability of different resources is also considered, including operating rooms and human resources such as surgeons. The problem assigns patients on the waiting list of surgical operations to operating rooms, determines the sequence of surgical operations and schedules the operations, simultaneously. Moreover, the sequence-dependent operating room preparation time and the time window constraint associated with each surgeon are considered in the problem. A genetic algorithm is developed to solve the problem inspired from the role model concept in sociology using simulating and differentiating procedures, namely Role Model Genetic Algorithm (RMGA).

In the next section, the literature on operating room planning and scheduling is reviewed. The problem assumptions and the mathematical model are presented in the third section. A new genetic algorithm is proposed in the fourth section to solve the problem. In the fifth section, the performance of RMGA is examined. Finally, the conclusions and suggestions for future studies are provided in the sixth section.

2-Literature review

In recent years, several comprehensive and precise reviews presented in the field of operating room planning and scheduling by Zhu et al. (2019), Gartner and Padman (2017),Cardoen, Demeulemeester, and Beliën (2010). As stated by Cardoen et al. (2010) decisions related to operating room planning and scheduling from a managerial perspective can be divided into three levels: strategic levels for the long term, tactical level for the medium term, and the operational level for the short term. The focus of this article is on the operational level. According to (Mateus, Marques, & Captivo, 2018) the operational level can be divided into two problems: advance scheduling and allocation scheduling. Breuer, Lahrichi, Clark, and Benneyan (2020) and Lin and Chou (2020) only investigate allocation scheduling during horizon planning. While in the advanced scheduling, patient are assigned to a day or an operating room, in allocation scheduling, They are sequenced or assigned to a specific time during the day. In this paper, both advanced scheduling and allocation scheduling are examined simultaneously in the same problem, as opposed to the studies conducted by Fei, Meskens, and Chu (2010), Jebali, Alouane, and Ladet (2006), Saadouli, Jerbi, Dammak, Masmoudi, and Bouaziz (2015), Landa, Aringhieri, Soriano, Tànfani, and Testi (2016) in which the two problems have been studied hierarchically and in two sequential steps. Dividing the operational level into two steps reduces the complexity of the problem (Molina, Fernandez-Viagas, & Framinan, 2015; Riise & Burke, 2011).However, due to the interdependence between these two sub-problems, the quality of the obtained surgical schedule decreases (Cardoen et al., 2010; Dios, Molina-Pariente, Fernandez-Viagas, Andrade-Pineda, & Framinan, 2015). Investigating the operating room planning and scheduling problem in an integrated manner and considering the nature of uncertainty during surgery in the problem, which is important for having an efficient scheduling process, is considered in few studies such as Kamran, Karimi, and Dellaert (2020), Roshanaei, Booth, Aleman, Urbach, and Beck (2020), Zhu, Fan, Liu, Yang, and Pardalos (2020), Akbarzadeh, Moslehi, Reisi-Nafchi, and Maenhout (2020), Dios et al. (2015), Saremi, Jula, ElMekkawy, and Wang (2013), Moosavi and Ebrahimnejad (2018), Molina et al. (2015), Díaz-López et al. (2018), Batun, Denton, Huschka, and Schaefer (2011), Addis, Carello, Grosso, and Tànfani (2016).

Kamran et al. (2020) investigated the problem with a modified block scheduling policy. They presented a mixed-integer linear programming model with multiple functions. Their developed solution approach is included a column-generation-based heuristic algorithm and Bender’s decomposition technique. The proposed method is dynamically able to manage unanticipated arrivals of emergency patients and tackle the disruptions in certain time blocks. Roshanaei et al. (2020) addressed a novel branch-and-check method for multi-level operating room planning and scheduling that solves the problem by a combination of constraint programming and integer programming. Two objective functions are considered: maximizing OR utilization and the number of surgeries. ORs cannot be shared between specialties and surgery durations are deterministic. Zhu et al. (2020) solve the problem of assigning different OR days to each specialty, allocation of operating rooms to surgeons and, patient assignment and sequence problems simultaneously. A hybrid GWO-VNS algorithm is developed to construct a good solution. The efficiency of their algorithm is proved by computational results and comparison whit other mainstream algorithms. Akbarzadeh et al. (2020) apply a column generation and driving heuristic to find a high-quality feasible solution at the operational level based on the expected demand. The problem is tested by a diverse artificial data and real data. The nurse re-rostering and nurse assignment to specific surgeries to maximize OR utilization is considered. Breuer et al. (2020) offer a robust optimization model. They considered the uncertainty of surgery duration, staff availability and anticipated of arrivals urgent and emergency patients in their model. Model performance is examined by real data. Díaz-López et al. (2018) developed a simulation-optimization technique to solve the surgical scheduling problem. Monte Carlo simulations were utilized for the probable duration of surgery and the meta-heuristic algorithm were used for optimization purposes. Moosavi and Ebrahimnejad (2018) presented a multi-objective mathematical model for scheduling elective patients, considering the intensive care unit and the patient's bed availability before and after surgery. A scenario was examined in the form of sustainable optimization in order to consider some probable parameters such as the duration of surgeries, the length of stay in the intensive care unit and the hospital room before and after surgery, and the demands of emergency patients. They proposed a new mixed-integer model based on the neighborhood search algorithm. Two sensitivity analyses were performed, one to determine the effect of the increase in the number of intervals on process time and the other to determine how increasing the limited resources of upstream and downstream units will affect scheduling. Addis et al. (2016) proposed a rotating horizon approach for patient selection and assignment. They considered scheduling and rescheduling steps simultaneously in their study. An integer linear programming model was repeatedly implemented to improve uncertainties between periods and minimize the waiting time and delays in patient allocation. Arranging patients to keep the number of cancellations caused by the arrival of new patients constrained and cancellation of patients were also considered. Moreover, to limit the number of cancellations due to the probabilistic duration of surgeries, they proposed a robust model for the linear integer problem. Batun et al. (2011) proposed a probabilistic two-stage mixed-integer model aiming to minimize operating room setup fixed costs, overtime costs, and surgeons' idle-time costs. Since their model was solved by standard probabilistic techniques at a reasonable time in real-world examples, they utilized the structural features of the model for computational improvements. Moreover, a set of valid inequalities were used within the L-shaped algorithm. Also, the L-shaped algorithm was used in the framework of the branch and cut algorithm, which makes it possible to solve real-world problems. Certain requirements in the surgical process were also considered. The resources involved in an operating room have a variety of synchronous characteristics and limitations. A complete surgery usually requires several resources. Time window constraints mainly affect resource allocation (Xiang, Yin, & Lim, 2015). Riise, Mannino, and Burke (2016) proposed a new generalized model for surgical scheduling problems. They showed how this model developed the project scheduling problem with multi-state resource constraints, multiple schemes, and typical time constraints. Then, they provided a search method to solve the proposed model. Xiang et al. (2015) presented the problem of surgical scheduling as an extension of the flexible job-shop scheduling problem with multiple resource constraints, which was then solved using an ant colony optimization algorithm with a two-tier sequential graph to integrate sequencing and resource allocation. Dios et al. (2015)proposed a decision support system for operating room scheduling. The system is capable of providing a short-term schedule, a medium-term evaluation, and manual modifications. They formulated their problem as a complex integer programming model and then solved it by approximate methods. Hashemi Doulabi, Rousseau, and Pesant (2016) and Doulabi, Rousseau, and Pesant (2014) developed an efficient column generation method for planning and scheduling operating rooms. Marques, Captivo, and Pato (2012) proposed a mixed-integer mathematical model to maximize the effective use of available resources. Non-optimal solutions are enhanced using improvement heuristics. In 2015, Marques et al. specifically designed improvement and constructive heuristics for the same problem. The heuristics presented in (Marques et al., 2015) were adapted for being employed in the genetic algorithm proposed in 2014 by Marques et al. They considered two inconsistent optimization criteria: 1) maximizing the use of the operating room 2) maximizing the number of scheduled surgeries, and they achieved better results than the ones obtained in their previous studies on the same problem. Marques, Captivo, and Pato (2015) used the minimization of a weighted Chebyshev distance from a return point to find efficient solutions to the multivariate optimization problem. Improvement and constructive heuristics are specifically designed to represent the objectives of the developed problem. Marques and Captivo (2015) continued their previous research by stating that their method (Marques et al., 2015) could only obtain one solution per time according to the weights assigned to each criterion which depends on decision-makers' priorities.

Then, they presented an evolutionary algorithm for the elective patients scheduling problem considering both tactical and operational levels. When a surgical operation is completed, the operating room preparation procedure must be performed before the next surgery is started, including operating room cleaning, providing the proper equipment, refilling disinfection supplies, and preparing the required staff (surgeons, nurses, and anesthesiologists) (Zhao & Li, 2014). Preparation times usually depend on the type of the two consecutive surgeries and are therefore sequence-dependent (Arnaout, 2010).When two consecutive surgeries belong to different types, more preparation time is required than when both consecutive surgeries are of the same type (Ciavotta, Dellino, Meloni, & Pranzo, 2010). Few studies have included sequence-dependent preparation times in their studies. Zhao and Li (2014) discussed three aspects of daily scheduling decisions in an outpatient surgery center, including the number of operating rooms to be opened, the allocation of surgeries to the operating rooms, and the sequence of surgeries in each operating room. They provided a nonlinear mixed-integer programming model and a constraint programming model to solve the problem aiming to minimize operating rooms' fixed costs overtime costs. They examined their problem under definite circumstances without considering the surgeon's time window constraint. Latorre-Núñez et al. (2016) proposed an integer linear programming model and converted it to constraint programming to solve the allocation sub-problem at the operational level. In order to solve large problems, they developed a metaheuristic based on a genetic algorithm and a constructive heuristic. Only in the studies conducted by Saremi et al. (2013) and Molina et al. (2015) both time window constraints and surgery duration uncertainty were simultaneously considered. However, the preparation time was not sequence-dependent in their study.

Molina et al. (2015) addressed operating room planning and scheduling considering surgical teams composed of one or two surgeons (house surgeon and assistant surgeon), in which the duration of surgeries depends on the surgical team's experience. They proposed a mixed-integer programming model and an extension of the multi-state block construction job-shop model (Pham & Klinkert, 2008) to solve it. Saremi et al. (2013) proposed three simulation methods based on optimization. Each patient moves along the preoperative, intraoperative, and postoperative stages. The process times of all three stages are uncertain. They examined different time distributions for different patients. The first method combines event-based simulation with tabu search to schedule surgeries. The second method is the integer program added to tabu search. The third method uses a 0-1 planning model combined with a heuristic and a simulation, based on tabu search to solve the problem.

Various solutions and methods have been presented for solving operating room planning and scheduling problems according to the nature of the problem (size and complexity). Zhao and Li (2014) describes them as exact algorithms and innovative algorithms such as genetic algorithms (Ala et al., 2020; Conforti et al., 2010; Fei et al., 2010; Guido & Conforti, 2017; Latorre-Núñez et al., 2016; Lin & Chou, 2020; Marques et al., 2014; Benoit Roland, Di Martinelly, & Riane, 2006; Benoît Roland et al., 2010), approximate methods (Dios et al., 2015), simulation (Díaz-López et al., 2018; Marques et al., 2012; Saadouli et al., 2015; Saremi et al., 2013) and Marco's decision-making process. Lin and Chou (2020) studied an operating room scheduling problem whit considering multifunctional operating rooms. They introduced heuristics to obtain feasible solutions and mentioned four local search procedures to improve their solutions. Finally, a hybrid genetic algorithm consisting of initial solutions, local search procedures ,and elite search procedure is used to the reported problem. Conforti et al. (2010) presented a multi-objective mathematical planning model by which the allocation of time intervals to surgical teams and the schedule of hospitalized patients are determined, aiming to maximize the use of operating rooms, maximize the number of scheduled patients, preferences of surgical teams and minimize the surgeons' idle time. In order to find the Pareto front, they proposed a metaheuristic approach based on the efficient execution of genetic algorithms. Ala et al. (2020) addressed a hybrid genetic algorithm to the appointment scheduling in the health care system to reduce the examination and preparation time of patients in rooms. Guido and Conforti (2017) defined a multi-objective integer linear programming model to integrate executive and operational decision-making levels. They extended the research conducted by Conforti et al. (2010) by defining schedules which are trade-offs between operating rooms' idle time, balanced distribution of operating rooms' active time between surgical teams, waiting time for surgeries, and overtime hours. The Pareto front guarantees that no criterion may be improved without the deterioration of another. To find non-dominant Pareto solutions, they proposed a hybrid approach based on the genetic algorithm. The proposed approach focuses on a method of initiation, which aims to build an initial population with an appropriate set of feasible chromosomes, which are the seeds for the evolution of the computational process. Benoît Roland et al. (2010) and Benoit Roland et al. (2006) developed a resource-constrained project scheduling model with emphasizing on the availability of human resources to address planning over several days and scheduling in one day with aims to minimize operating rooms' setup and overtime costs and proposed a genetic algorithm to solve it. Researchers such as Lin and Chou (2020), Benoît Roland et al. (2010), Benoit Roland et al. (2006) and Conforti et al. (2010) investigated their problem under definite conditions without considering the time window constraints and sequence-dependent preparation times. In this paper, a nonlinear mixed-integer mathematical model and a new genetic algorithm are proposed to solve a different combination of operating room planning and scheduling problems at the operational level, considering the sequence-dependent preparation times and time window constraints. Table (1) and table (2) summarize the studies conducted in the field of scheduling and planning of operating rooms from different aspects. In table 1, there are 5 patient types as follows: 1:outpatient , 2: inpatient, 3: emergency patient, 4: necessary patient, and 5 :unspecified.

More over, there are 15 Objective functions in this table as follows: 1: patient waiting time, 2: surgery cancellations, 3: OR utilization, 4: completion time, 5: surgeon idle time, 6: cost, 7: operating room overtime, 8: OR idle time, 9: operating room setup, 10: hospitalization, 11: number of scheduled patients, 12: number of beds, 13: surgeon overtime, 14: delay, 15: other.

Additionally 19 solution methods are considered in this table as follows: 1: mathematical programming model, 2: constraint programming, 3: goal programming, 4: column generation, 5: robust optimization, 6: genetic algorithm, 7: discrete event simulation, 8: branch and price, 9: tabu search, 10: neighborhood search technique, 11: quadratic programming, 12: Monto Carlo, 13: greedy algorithm, 14: branch and price and cut, 15: evolutionary algorithm, 16: constructive and improving heuristics, 17: approximate Methods, 18: metaheuristic, 19: other heuristics.

Finally, 10 source types are presented in table 1 as follows: 1: surgeon, 2: operating room, 3: nurse, 4: anesthesiologist, 5: recovery beds, 6: ICU beds, 7: ward beds, 8: equipment, 9: instruments, 10: other.

Table 1. Research classifications in the literature

Authors	Type of patient	Objective function	Solution method	Resources type
Kamran et al. (2020)	1-2-3	1-2-5-7-14	1-4-19	1-2-5-7
Roshanaei et al. (2020)	1-2	3-11	1-19	1-2
Lin and Chou (2020)	1-2	3-6-7-8	1-6-10-19	1-2
Zhu et al. (2020)	1-2	1-6-7	1-10-18-19	1-2
Breuer et al. (2020)	1-2-3-4	1-3-7-15	1-5	1-2-3-4
Akbarzadeh et al. (2020)	1-2	6-15	1-4-19	2
Ala, Torkayesh, Torkayesh, and Iranizad (2020)	1-2	15	1-6	1
Molina-Pariente, Hans, and Framinan (2018)	1-2-3	2-6-7-8	12-13-17	1-2
Moosavi and Ebrahimnejad (2018)	1-2-3	1-6-7-8-12-15	1-5-10	2-6-7
Mateus et al. (2018)	1-2	11-15	1-10-19	1-2
Latorre-Núñez et al. (2016)	1-2-3	4	6-16	1-3-4-5-8-9-10
Díaz-López et al. (2018)	5	3-14	1-12-13	2
Dios et al. (2015)	1-2	5-11-14	1-17	1-2
Landa et al. (2016)	1-2	2-3	1-10-12-13	2-7
Conforti et al. (2010)	2	3-11-15	1-6	1-2
Van Huele and Vanhoucke (2014)	1-2	6-7	1	1-2-5-6
Hashemi Doulabi et al. (2016)	1-2	15	1-2-4-14	1-2
Doulabi et al. (2014)	1-2	15	1-2-4	1-2
Batun et al. (2011)	5	5-6-7-9	1	1-2
Marques et al. (2015)	1-2	3-11	1-16	1-2
Marques, Captivo, and Pato (2014)	1-2	3-11	1-6	1-2
Marques et al. (2012)	1-2	3	1	1
Marques et al. (2015)	1-2	3-11	15	1-2
Addis et al. (2016)	1-2-3	1-14	1-5	---
Aringhieri, Landa, Soriano, Tànfani, and Testi (2015)	2	1-6-12	1-18	1-2-7
Vijayakumar, Parikh, Scott, Barnes, and Gallimore (2013)	1-2	11	1-19	1-2-3-8
Riise and Burke (2011)	1-2	1-13-15	1-18	1-2
Durán, Rey, and Wolff (2017)	5	15	1-16	1-2
Riise et al. (2016)	1-2	1-6-15	1-18	1-2-3-4-5-6-7-8
Benoît Roland, Di Martinelly, Riane, and Pochet (2010)	1-2	6-7-9	1-6	1-2-3-4-10
Zhao and Li (2014)	1	6-7-9	1-2	---
Saadouli et al. (2015)	1-2	1-4-7-8	1-7	2-5
Guido and Conforti (2017)	1-2	6-7-8-11-15	1-6	1-2
Xiang et al. (2015)	1-2	4	1-7-18	1-2-3-4-5-6-7
Ghazalbash, Sepehri, Shadpour, and Atighehchian (2012)	1-2	4-8	1	1-2-3-4-7-8-9-10
Saremi et al. (2013)	1	1-2-4	1-7-9	1-2-3-5-7
Agnetis et al. (2014)	1-2	6-15	1	1-2
Dios et al. (2015)	1-2	15	1-17	1-2
Current research	1-2	5-6-7-8-9	1-6	1-2

Table 2. Problem specifications in the literature

Authors	Type of scheduling		Decision levels		Environmental factors																	
			uncertainties						cancellations	No show				emerge		preferences		ORs			Turnover time	
					operational				Duration of stay										multiple			
	Block scheduling	Open scheduling	strategic	tactical	Advanced scheduling	Allocation scheduling	Duration of surgery	Patient arrival	upstream	Down stream	delay	interruptions	necessary patient		Time-window constraint	patient	surgeon	single	identical	Non-identical	OR	surgeon
(Kamran et al., 2020)	✓				✓	✓	✓	✓	✓	✓	✓		✓		✓					✓	✓	✓
(Roshanaei et al., 2020)		✓			✓	✓													✓		✓	
(Zhu et al., 2020)	✓			✓	✓	✓	✓												✓			
(Lin & Chou, 2020)		✓				✓													✓			
(Breuer et al., 2020)	✓					✓	✓	✓						✓	✓		✓		✓			
(Akbarzadeh et al., 2020)	✓				✓	✓	✓										✓		✓			
(Ala et al., 2020)		✓			✓	✓														✓		
(Moosavi & Ebrahimnejad, 2018)	✓				✓	✓	✓		✓	✓					✓					✓	✓	
(Díaz-López et al., 2018)					✓	✓	✓												✓			
(Mateus et al., 2018)	✓				✓															✓	✓	
(Saadouli et al., 2015)		✓			✓	✓	✓			✓									✓		✓	
(Guido & Conforti, 2017)	✓			✓	✓	✓														✓		
(Molina, Hans, & Framinan, 2018)		✓			✓		✓				✓				✓				✓			
(Molina et al., 2015)		✓			✓	✓	✓													✓		
(Latorre-Núñez et al., 2016)						✓	✓								✓					✓	✓	✓
(Hashemi Doulabi et al., 2016)		✓			✓	✓													✓		✓	
(Doulabi et al., 2014)		✓			✓	✓													✓		✓	
(Zhao & Li, 2014)		✓			✓	✓														✓	✓	
(Landa et al., 2016)	✓				✓	✓	✓				✓								✓		✓	
(Batun et al., 2011)	✓				✓	✓	✓		✓	✓									✓		✓	✓

Table 2. Continued

Authors	Type of scheduling		Decision levels		Environmental factors												preferences		ORs			Turnover time	
					uncertainties						cancella tions	No show											
					operational				Duration of stay											multiple			
	Block scheduling	Open scheduling	strategic	tactical	Advanced scheduling	Allocation scheduling	Duration of surgery	Patient arrival	upstream	Down stream	delay	interruptions	necessary patient	emerge	Time-window constraint	patient	surgeon	single	identical	Non-identical	OR	surgeon
(Marques et al., 2012)		✓			✓	✓													✓		✓	
(Marques et al., 2014)		✓			✓	✓													✓		✓	
(Marques et al., 2015)		✓			✓	✓													✓		✓	
(Marques & Captivo, 2015)	✓			✓	✓	✓													✓		✓	
(Van Huele & Vanhoucke, 2014)		✓		✓	✓	✓											✓			✓		
(Aringhieri et al., 2015)	✓			✓	✓														✓			
(Fei et al., 2010)		✓			✓	✓													✓		✓	
(Xiang et al., 2015)		✓			✓	✓													✓			
(Vijayakumar et al., 2013)	✓				✓	✓							✓						✓			
(Durán et al., 2017)	✓				✓	✓														✓		
(Pham & Klinkert, 2008)		✓			✓	✓									✓					✓	✓	
(Dios et al., 2015)					✓	✓					✓		✓							✓		
(Riise & Burke, 2011)	✓			✓	✓	✓							✓									
(Ghazalbash et al., 2012)		✓			✓	✓														✓	✓	
(Conforti et al., 2010)	✓			✓	✓	✓													✓			
(Riise et al., 2016)					✓	✓														✓	✓	
(Addis et al., 2016)	✓				✓	✓	✓				✓		✓	✓	✓				✓			
(Benoît Roland et al., 2010)	✓				✓	✓														✓		
(Saremi et al., 2013)					✓	✓	✓	✓	✓	✓												
(Agnetis et al., 2014)	✓			✓	✓															✓		
Current research		✓			✓	✓	✓													✓	✓	

3-Problem definition

In this section the problem specifications is described. It is assumed that there is a waiting list of elective patients, in which the type of each surgery and the surgeon of each patient are specified. Patients and surgeons should be scheduled and planned so that they are available at the hospital at the specified time. Therefore, when allocating a surgery to a time and an operating room, the surgeon's availability and the time availability in the allowed operating room should be considered to perform the surgery. The integration of the allocation of surgeries to operating rooms and their sequence in each operating room is reviewed on a daily short-term horizon. In this problem, it is intended to determine the following decisions:

- The assignment of surgeries to operating rooms;
- The priority of surgeries assigned to each operating room;
- The starting time of each surgery;
- The state of each operating room (active or inactive).

3-1- Problem assumptions

The assumptions of the problem in this study are as follows:

- All resources are always available except operating rooms and surgeons.
- The time required for cleaning and the preparation of operating rooms is calculated separately and is added to the duration of the operation.
- The duration of operating rooms’ cleaning and preparation depends on the sequence of surgeries in each operating room.
- Operating rooms are not identical. Therefore, some types of surgeries are not allowed to be performed in some operating rooms.
- Emergency patients are not considered in the problem because patients admitted to the emergency department usually undergo surgery immediately. Hence, only elective patients are considered in this study.
- Surgeries cannot be stopped once they are started.
- Two surgeries can't overlap in an operating room.
- A surgeon cannot be present in two operating rooms at the same time.
- The time horizon for the operating rooms’ planning and scheduling is daily.
- The duration of surgery is a random variable.
- Each operating room has a certain amount of overtime, which is applied as a penalty in the objective function.
- There is a time window constraint for each surgeon, which indicates the surgeon's presence time in the hospital.

Figure (1) shows an example, in which 7 surgeries are scheduled in 3 operating rooms. Note that in operating room 3 (OR3) after surgery 5 is done (P5) and the operating room is cleaned and prepared for the next surgery, surgery 3 (P3) is delayed because surgeon 1 (S1) is unavailable, being busy performing surgery 2 (P2) in operation room 2 (OR2), making the operating room idle.

Surgeon 3 (S3) should wait for the surgery on patient 6 to end (P6) as operating room 1 is unavailable (performing surgery 4 (P4) by surgeon 1 (S1) and cleaning and preparing the operating room) until time t' to for the next surgical procedure. This will cause the surgeon to become idle. Also, surgeon 2 (S2) should wait after completing their initial surgery in operation room 3 (OR3) as operation 2 is occupied. There is no need for a surgeon to be present in the operating room when preparing and cleaning the operating room. Each operating room is allowed a certain amount of overtime, and all surgeries performed by a surgeon must be performed during the surgeon's presence in the hospital.

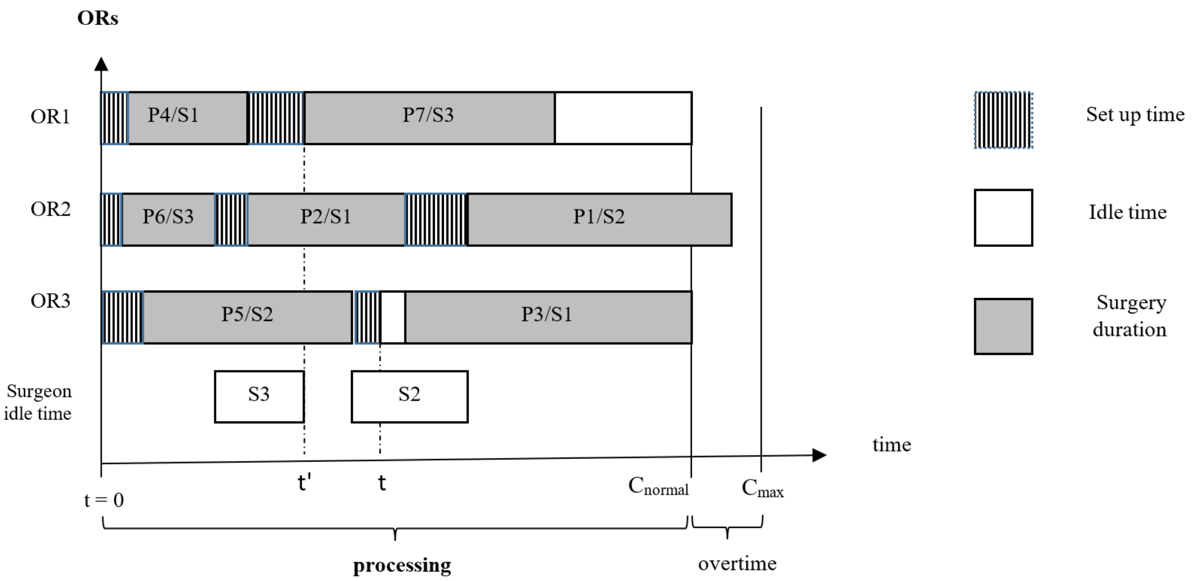

Fig 1. Surgery scheduling process

3-2- Mathematical model of the problem

In this section, a mixed-integer nonlinear mathematical model is presented, as an extension of the model proposed by Zhao and Li (2014). The parameters of the problem are described in table (3).

Table 3. Sets, parameters, and variables for the mathematical model

Sets and Indices:

R: Total number of available operating rooms

i: Operating rooms index

P: Total number of surgeries which should be scheduled

j: Surgeries index

S: Set of Surgeons

N: Total number of surgery types

n: Index of surgery type

Q: A large positive number

K: Location of a surgery in an operating room

Parameters:

$\mathbf{s_j}$: surgeon of patient i

$\mathbf{C_s^{max}}$: Maximum surgery completion time performed by surgeon s

$\mathbf{C_s^{idletime\ surgeon}}$: Cost of one idle time unit for surgeon s

$\mathbf{SD_j}$: Duration of surgery j

$\mathbf{t_{jj'}^{st}}$: Preparation Duration of for surgery j' when the previous surgery was j

$\mathbf{t_i^{tot}}$: Normal availability time of operating room i

$\boldsymbol{C_i^{overtime}}$: Overtime Cost of operating room i for each overtime unit

$\boldsymbol{C_i^{idletime}}$: Idle cost of operating room i per unit of idle time

$\boldsymbol{CO_i}$: Setup cost of operating room i

$\boldsymbol{MAX\ T_i^{OR}}$: Maximum time that operating room i can be open

$\boldsymbol{C_j}$: Completion time of surgery j

$\boldsymbol{T_{max}^s}$: Maximum time allowed for the surgeon to be available for performing surgeries

Subsets:

$\boldsymbol{P_s^S}$: Set of surgeries performed by surgeon s

decision and auxiliary variables:

I_s^{min}: Minimum start time of surgery s $Z_s^{idletime\ surgeon}$: Idle time between surgeries of surgeon s l_{jn}: Equals 1, if and only if surgery j is of type n, otherwise 0 h_{in}: Equals 1, If and only if surgery type n can be performed in operating room i, otherwise 0 X_{ijk}: Equals 1, If and only if surgery j is to be the kth surgery in operating room i, otherwise 0 Z_i: Period in which operating room i is overloaded F_i: Period in which operating room i is idle y_i: Equals 1, If and only if operating room i is open, otherwise 0 $y_{jj'}$: Equals 1, If and only if patient j is in the common source before patient j', otherwise 0

The mathematical model of the problem is as follows:

$$opt\ f(x) = MIN\ [f_1(x).f_2(x).f_3(x).f_4(x)]$$

$$f_1 = \sum_{j=1}^{P} y_i (CO_i + C_i^{overtime} Z_i) \quad (1)$$

$$f_2 = \sum_{j=1}^{P} y_i C_i^{idletime} F_i \quad (2)$$

$$f_3 = C_{max} \quad (3)$$

$$f_4 = C_s^{idletime\ surgeon} \sum_{s=1}^{S} Z_s^{idletime\ surgeon} \quad (4)$$

$$\sum_{k=1}^{P} X_{ijk} \leq \sum_{n=1}^{N} h_{in} l_{jn} \quad \forall i.j \quad (5)$$

$$\sum_{i=1}^{R} \sum_{k=1}^{P} X_{ijk} = 1 \quad \forall j \quad (6)$$

$$\sum_{j=1}^{P} X_{ij(k+1)} \leq \sum_{j=1}^{P} X_{ijk} \quad k = 1 \dots P-1. \forall i \quad (7)$$

$$C_j + Q * \left(2 + y_{jj'} - \sum_{k=1}^{P} X_{ijk} - \sum_{k=1}^{P} X_{ij'k}\right) \geq C_{j'} + SD_j$$
$$C_{j'} + Q * \left(3 - y_{jj'} - \sum_{k=1}^{P} X_{ijk} - \sum_{k=1}^{P} X_{ij'k}\right) \geq C_j + SD_{j'} \quad (8)$$

$$\forall s \in S. j \in P_s^S. j' \in P_s^S. j \neq j'. j < j'$$

$$C_j + Q * \left(2 + y_{jj'} - \sum_{i \in R} \sum_{k=1}^{P} X_{ijk} - \sum_{i' \in R} \sum_{k=1}^{P} X_{i'j'k}\right) \geq C_{j'} + SD_j \quad (9)$$

$$C_{j'} + Q * \left(3 - y_{jj'} - \sum_{i\in R}\sum_{k=1}^{P} X_{ijk} - \sum_{i'\in R}\sum_{k=1}^{P} X_{i'j'k}\right) \geq C_j + SD_{j'}$$

$$\forall s \in S . j \in P_s^S . j' \in P_s^S . j \neq j' . j < j' . s_j = s_{j'} = 1$$

$$C_j \geq \sum_{i=1}^{R}\sum_{k=1}^{P}\left(t_{j'j}^{st} + SD_j\right)X_{ijk} \quad \forall j \in P \tag{10}$$

$$\sum_{j\epsilon P_s^S}\sum_{i=1}^{R}\sum_{k=1}^{P} SD_j\ X_{ijk} \leq T_{max}^{s} \quad \forall s \in S \tag{11}$$

$$I_s^{min} \leq C_j - \sum_{i=1}^{R}\sum_{k=1}^{P} SD_j\, X_{ijk} \quad \forall s \in S . \forall j \in P_s^S \tag{12}$$

$$C_s^{max} \geq C_j \qquad \forall s \in S . \forall j \in P_s^S \tag{13}$$

$$Z_s^{idletime\ surgeon} \geq C_s^{max} - I_s^{min} - \sum_{j\in P_s^S}\sum_{i=1}^{R}\sum_{k=1}^{P} SD_j\, X_{ijk} \quad \forall s \in S \tag{14}$$

$$C_j \leq \sum_{i=1}^{R}\sum_{k=1}^{P} t_i^{tot}\, X_{ijk} \quad \forall j \in P \tag{15}$$

$$C_{max} \geq C_j \qquad \forall j \in P \tag{16}$$

$$\sum_{j=1}^{P} X_{ij1} \leq y_i \quad \forall i \tag{17}$$

$$\sum_{j=1}^{P}\sum_{k=1}^{P} X_{ijk}\, SD_j + \sum_{j=1}^{P}\sum_{k=1}^{P-1}\sum_{j'=1}^{P} X_{ijk}\, X_{ij'(k+1)}\, t_{jj'}^{st} - t_i^{tot} \leq Z_i \quad \forall i \tag{18}$$

(19)

$$t_i^{tot} - \left(\sum_{j=1}^{P}\sum_{k=1}^{P} X_{ijk}\, SD_j + \sum_{j=1}^{P}\sum_{k=1}^{P-1}\sum_{j'=1}^{P} X_{ijk}\, X_{ij'(k+1)}\, t_{jj'}^{st}\right) \leq F_i \ \forall i \tag{20}$$

$$F_i \geq 0 \quad \forall i \tag{21}$$

$$MAX\, T_i^{OR} \geq t_i^{tot} + Z_i \quad \forall i \tag{22}$$

$$X_{ijk} \in \{0،1\} \ \forall i . j . k \tag{23}$$

$$Y_{jj'} \in \{0،1\} \ \forall j . j' \in J\, ; j \neq j' \tag{24}$$

(25)

$$W_{jj's} \in \{0,1\}\ \forall s \in S. \forall j.j' \in J\,; j \neq j'$$

$$Z_s^{idletime\ surgeon}.C_s^{max}.C_j.I_s^{min} \geq 0\ J\,; j \neq j' \tag{26}$$

Objective functions (1) and (2) minimize setup costs, operating room overtime costs, and operating room idle-time costs, respectively. Objective function (3) minimizes the closing time of the last operating room in use. Objective function (4) minimizes surgeons' free time window. Constraint set 5 ensures that surgeries are performed in allowed operating rooms. If n-type surgeries cannot be performed in operation i (hin = 0), then $Xijk$ variables for all surgeries j with $ljn=1$ are equal to be zero. Constraint set 6 indicates that each surgery must be scheduled exactly once. Constraint set 7 states that surgery K + 1 can be scheduled only when the previous K surgeries are scheduled in that operating room. Constraint set 8 states that each operating room cannot accommodate more than one patient at a time. Constraint set 9 defines the surgery priorities for surgeon s to prevent them from overlapping. constraint set 10 ensures that the completion time of each surgery in each operating room, is greater than the sum of the time of surgery and its preparation time. Constraint set 11 applies the limitation of surgeons' daily working time. Constraint sets 12 and 13 define the earliest start time and the latest completion time for each surgeon during the day, respectively. Constraint set 14 limits each surgeon's waiting time per day. Constraint set 15 states that the time required for completing surgery should be less than or equal to the normal available time in the operating room. Constraint set 16 specifies that the maximum completion time is greater than or equal to the closing time of all operating rooms. Constraint set 17 states that surgeries can only be assigned to open operating rooms. Constraint sets 18 and 19 define the amount of overtime in each operating room. Since the goal is to minimize the costs of operating rooms' overtime, if operating room i is not open longer than normal, then Zi equals zero. If the operating room is open longer than normal, then Zi equals the overtime. Constraint sets of 20 and 21 define the idle time of the operating room. Constraint set 22 defines the maximum time available for the operating room. Constraint sets 23, 24, and 25 indicate that the decision variables $w_{jj's}.X_{ijk}.Y_{jj'}$ are binary. Constraint set 26 defines the ranges of variables.

4-Solution method

Due to the NP-hard nature of the problem, exact methods are not capable of finding the optimal solution in a reasonable time, and heuristic or meta-heuristic methods must be used to solve the problem to find the best possible (near-optimal) solution in a reasonable time. In this section, a new genetic algorithm based on the role model concept in sociology is presented to solve the operating rooms' planning and scheduling problem. The genetic algorithm deals with a population consisting of a large number of individuals and evolves the population using a set of specific selection rules to maximize the fitness value (in other words, minimize the cost function). In the genetic algorithm, each potential solution is considered as a chromosome in the initial population and the corresponding fitness function is assigned to it. In each generation, new chromosomes are produced by different genetic operators, namely mutation and crossover. The chromosomes of the next generation are selected from the previous generation using a criterion according to their fitness function. This procedure is repeated until the termination condition is satisfied.

RMGA it is inspired from the role model concept proposed by sociologist Robert K. Merton (Merton, 1957). In RMGA two sets of good and bad role models are considered. The good role model set consists of a number chromosomes having better fitness values than other chromosomes. On the other hand, the bad role model set consists of a number chromosomes having worse fitness values than other chromosomes. People in the society try to imitate from the good role models and be different from bad role models. In the mutation operator of RMGA, a randomly selected chromosome from the population tries to imitate some properties from a randomly selected good role model and be different from a randomly selected bad role model. Details of RMGA is described as follows:

4-1- Chromosome structure

The chromosome structure in RMGA is two dimensional. The vertical dimension represents the operating rooms and the horizontal dimension represents the surgeries assigned to each operating room.

For each operating room, there is an array whose length and arrangement of elements indicate the number and order of surgeries assigned to that operating room. If the number of surgeries assigned to operating rooms increases or decreases, the length of the corresponding string will also increase or decrease. In RMGA, the structure of the chromosome is two-dimensional and the strings' lengths are variable. For further elaboration, suppose there are 10 patients and 3 operating rooms. In this case, figure (2) shows the structure of the chromosome of a solution and its decoding process. The first string consists of a random arrangement of integers from 1 to N (number of patients), and the second string specifies the allowed operating room allocated to the patients in the first row. Each patient in the first row is assigned according to the operating room specified in the second row. The patients' sequence is determined based on the random order created in the first row of the matrix.

Patients	10	5	9	1	6	8	7	3	4	2
Operating room	1	1	3	1	3	3	2	3	2	3

$$\begin{array}{c} OR1 \\ OR2 \\ OR3 \end{array}\begin{bmatrix} 10 & 5 & 1 & 0 & 0 \\ 7 & 4 & 0 & 0 & 0 \\ 9 & 6 & 8 & 3 & 2 \end{bmatrix}$$

Fig 2. A feasible chromosome and its decoding process.

4-2- Crossover and mutation

Both simulation and differentiation processes are used in the crossover and mutation operations.
Simulation process: In this process, one chromosome is named as influencer and the other as follower. One or more patients are randomly selected and their assignments in the follower chromosome are changed to be same as the influencer chromosome.
Differentiation process: Like simulation process, there are influencer and follower chromosomes. But in the differentiation process, one or more patients are randomly selected from the follower chromosome and their assignments are checked to be different from their position in the influencer chromosome. If an assignment is similar in both chromosome, then it should be changed to a randomly different feasible position. If the selected patient is only allowed to be assigned to one position and that position needs to be changed according to the differentiation process, then the differentiation process will not be performed for that patient.

The number of selected genes to perform a simulation or differentiation process is determined by a coefficient of the total number of genes of chromosomes, namely impact factor. For example, assume the impact factor and the number of patients (genes) to be 0.3 and 10, respectively. In this case 3 (10 * 0.3) genes should be selected to perform a simulation or differentiation process.

Y1= influencer chromosome

6	2	10	
9	1	4	
5	8	3	7

Y2 = follower chromosome

2	3	1		
9	7	5	6	8
10	4			

$\longrightarrow$ $Y2_{new}$ =

2	1			
9	7	5	6	8
3	4	10		

Fig 3. The crossover operation

The crossover and mutation operators in the proposed algorithm work as follows:
Crossover: People in a community are influenced by each other. Two chromosomes are randomly selected, one of them is considered as follower and the other is considered as influencer. Thereafter, the simulation process takes place between them.

Mutation: In a community, some people are known as good role models and some as bad role models. People tend to follow good role models and avoid bad ones. In the mutation operator, a chromosome is randomly selected as follower and a chromosome is selected from the good role models set called good influencer and a chromosome is selected from the bad role models set called bad influencer.
Consequently, the simulation process is occurred between the follower chromosome and the good influencer and the differentiation process is occurred between the follower chromosome and the bad role model chromosome. Consider the following example:
Suppose chromosome a is selected randomly, chromosome G is selected from the good role model set with the best fitness value, and chromosome B is selected from the bad role model set that have the worst fitness value. Also, the impact factor and the number of patients are 0.3 and 10, respectively. Therefore, 3 (10 * 0.3) genes must be selected randomly in chromosome a in order to be simulated based on the best chromosome, and then three other genes must be selected in order to be differentiated according to the worst chromosome. Suppose that the three genes selected for the simulation process are 6, 7, and 2, and the three genes selected for the differentiation process are 3, 10, and 8. Figure 4, shows the result of performing the mutation operator on chromosome C.

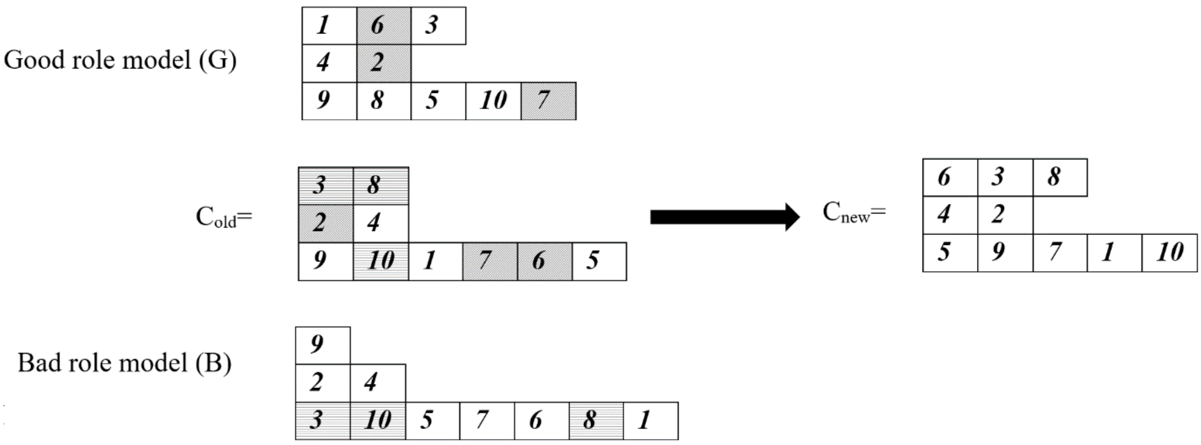

Fig 4. Perform the simulation and differentiation processes in the mutation operator

4-3- The proposed algorithm

The implementation of RMGA process is as follows:

Step 1- Generate the initial population: Create an initial population of random chromosomes. The population number is determined by a parameter named *npop*.

The chromosomes are produced as follows:

Step 1-1. Generate two strings, each containing N arrays. The arrays in the first string contain non-repetitive random integers numbers from 1 to N (number of patients). The value of each array in the second string is a random integer number from 1 to m (number of operating rooms).

Step 1-2. Calculate the objective function value for each chromosome. The chromosome’s objective function values are calculated as follows:

The cost function is composed of five stages:

1.2.1 Perform the first step of scheduling patients based on the allocation and priority specified in the corresponding chromosome (arrangement of patients in different operating rooms), the operating rooms’ preparation time, and the duration of surgery.

2.2.1 The first penalty function is calculated based on the difference between the maximum occupation time of the operating room and the minimum occupation time of the operating room, that is the difference between the occupancy periods of the two operating rooms.

3.2.1 The second penalty function is calculated based on the surgeon's unemployment time and is considered as the second cost function.

4.2.1 Third Objective Function: if two different surgeries are assigned to one surgery at the same time, the overlapping time intervals are calculated and for each minute of the surgeon's overlap time, a λ set of penalty units is assigned, so that this assignment is likely to be removed from selection priorities due to the high value of the cost function.

5.2.1 Fourth penalty function: if the duration of an operation exceeds the allowable time window of a surgeon, this assignment will probably either be removed by considering a β penalty for it, or be corrected in a way that it is performed within the allowable time window,

6.2.1 Fifth penalty function: If the duration of an operation exceeds the permitted working period of the operating room, it is considered unallowable by assigning a unit if ρ penalty. This operation will eventually be corrected in a way that it is performed within the permitted working period.

Note that depending on the type of surgery, the operating room assigned to each patient is necessarily correct and does not need to be controlled. Finally, the penalty functions of surgeons' idle time, exceeding surgeons' allowable time window, exceeding operating rooms permitted working period, and overlapping surgeries assigned to a surgeon are calculated and the summation of all cost functions is considered as the final penalty function.

Step 2. Create sets of good and bad chromosomes: select as much as n-good chromosomes with the best fitness function and n-bad chromosomes with the worst fitness as the good and bad chromosome sets, respectively.

Step 3. Perform the crossover operation: The number of crossover operations in each iteration is fixed and determined by a coefficient of *npop*, named cross-rate, which is one of the parameters of the genetic algorithm. Perform the crossover operation as follows:

Step 3-1. Determine the number of changeable genes using the impact factor: define a value as the impact factor according to the number of patients and call it α. Multiply the number of patients by the impact factor and name the result K.

Step 3-2. Select the chromosomes: Randomly select two chromosomes and name them Y1 and Y2.

Step 3-3. Simulation process between the two chromosomes Y1 and Y2: randomly select K genes in chromosome Y1. Simulate the selected genes, according to their location on chromosome Y1, in chromosome Y2. Then randomly select K genes on chromosome Y2. Simulate the selected genes, according to their location on chromosome Y2, in chromosome Y1.

Step 3-4. Calculate the fitness function and replace the chromosomes: Calculate the objective function value of the new chromosomes Y1 and Y2 and replace them with the old chromosomes.

Step 4. Perform the mutation operation: The number of mutation operations in each iteration is fixed and is determined by a coefficient of npop, named mut-rate, which is one of the parameters of the genetic algorithm. Perform the mutation operation as follows:

Step 4-1. Select a chromosome to perform the mutation operation, a good chromosome, and a bad chromosome: first, randomly select a chromosome from the initial population. Also, randomly select a chromosome from the set of good chromosomes and a chromosome from the set of bad chromosomes and name them C, CG, and CB, respectively.

Step 4-2. Simulate based on a good chromosome: select K genes randomly from chromosome C. Simulate the position assigned to these genes based on the best selected chromosome. At each stage of the simulation process, the sequence in each row is random.

Step 4-3. Differentiate based on a bad chromosome: select K genes randomly from chromosome C, and then differentiate the position assigned to these genes in the selected chromosome based on the worst selected chromosome. At each stage of the differentiation process, the sequence in each row is random.

Step 4-4. Calculate the fitness function and replace the chromosomes: Calculate the objective function value of the new C chromosome and replace the new chromosome with the previous one.

Step 5. Check the termination criterion: the termination condition is reaching to Max_itr generations.

Step 6. Update the set of good and bad chromosomes: if in the current population there are chromosomes with a better objective function value than the chromosomes in the good chromosomes set, place it in the set of good chromosomes and delete the worse chromosome in the set, do the same process to update the bad chromosome set.

We empirically found that values of 0.5 for cross-rate, 0.5 for mut-rate, 0.2 for α, 200 for the initial population, 200 for Max_itr, 5 for n-good, and 5 n-bad prepare good results in a reasonable time.

5-Numerical experiments

In this section, the proposed algorithm is compared to a conventional genetic algorithm namely CGA and a hybrid genetic algorithm proposed for the nearest problem in the literature (Ala et al., 2020), namely HGA. CGA has the same structure as RMGA, but in CGA there is not good and bad role model sets and the Exchange mutation operator is performed. On the other hand, the problem considered by (Ala et al., 2020) did not consider the surgeons in the scheduling procedure and only patients and operation rooms were considered in the scheduling procedure. While in the current problem, three parameters of surgeons, patients, and operation rooms are considered.

5-1- Generating random data for the problem

The problem investigated in this study has different parameters. We consider 3 parameters of the number of patients, the number of operating rooms, and the number of surgeons and consider 3 levels for each of them as shown in table 4. For other parameters only one level is considered.

Table 4 . Problem parameters

	Parameter	Level 1	Level 2	Level 3
1	The number of patients	20	30	40
2	The number of operating rooms	2	4	6
3	The number of surgeons	4	6	8
4	Number of surgery types		6	
5	Operation times		U[5,15]	
	Operation preparation times		U[10,50]	

By combining different levels of the parameters, 27 types of problems (3 * 3 * 3) are randomly generated. Six types of surgery are considered with different durations which are defined using the uniform distribution U[10,50]. The preparation times are also determined by the uniform distribution U[5,15]. If two consequent operations are identical, then a constant preparation time of 5 is considered. All computer programs used in this research are written by the MATLAB 2016 programming language and run by an Intel Core i7, 2GHz, and 6GB RAM computer.

5-2- Comparison between RMGA and CGA

The all 27 generated test problems based on table 4 are solved by both RMGA and CGA. Because of the random nature of genetic algorithms, each run may prepares a different result. Therefore, each problem is solved 20 times by each algorithm. The 27 randomly generated problems can be divided into three categories in terms of the number of patients. The first category consists of problems with 20 patients and the problems in the third and fourth categories have 30 and 40 patients, respectively. Each category consists of 9 problems, the average solutions obtained from solving them by each of the two algorithms are shown in table 5. Similarly, the problems were also divided into three groups of 9 in terms of the number of operating rooms and the number of surgeons, whose average solutions obtained from solving them by each of the two algorithms are shown in table 5. To evaluate the performance of the proposed algorithm, all 27 random problems are solved by the RMGA and CGA, and their results are presented in table 5, in terms of the different states shown in table 4. The average CPU time and the average results obtained by the CGA and the RMGA are shown in columns 5, 6, 7, and 8 of table 5. In columns 9, 10, 11, and 12, the number 1 indicates which algorithm has found a better maximum or minimum between different runs of the algorithms, respectively. RMGA obtained a better maximum value in 25 cases and a better minimum value in 24 cases than CGA. Additionally, table 5 shows the average results obtained by RMGA is better than CGA.

Table 5. Comparing the results of RMGA with the result of CGA

Problem number	Number of patients	Number of ORs	Number of surgeons	Mean Results of CGA	Mean CPU times of CGA	Mean Results of RMGA	Mean CPU times of RMGA	Max		Min	
								CGA	RMGA	CGA	RMGA
1	20	2	4	5421.2	27.92	4827	28.119	0	1	0	1
2			6	46605.3	33.926	45948.3	34.939	0	1	0	1
3			8	33317.8	35.21	33042	36.072	0	1	0	1
4		4	4	42987.5	31.756	42448.2	33.858	0	1	0	1
5			6	13907.6	32.038	12575.8	32.865	0	1	1	1
6			8	27161.3	31.994	26893	32.786	0	1	0	1
7		6	4	18194	33.759	16517	34.557	0	1	0	1
8			6	52267.4	34.414	40422.8	35.337	0	1	0	1
9			8	41108.4	36.94	41099.9	37.037	0	1	0	1
10	30	2	4	62299.2	33.1	61348.5	34.832	0	1	0	1
11			6	69282.6	34.265	58833.3	35.392	0	1	0	1
12			8	48880.2	34.65	47232.5	35.135	0	1	0	1
13		4	4	68161.5	39.13	67449.4	42.426	0	1	1	1
14			6	50153.6	41.43	48703	43.349	0	1	0	1
15			8	48743.5	41.57	44729.8	42.83	0	1	0	1
16		6	4	77290.3	42.34	77133.2	45.289	0	1	0	1
17			6	68022.8	43.51	67979.2	44.802	0	1	0	1
18			8	47703.8	46.643	47266.7	49.768	0	1	0	1
19	40	2	4	211293	37.58	209917.6	38.925	0	1	0	1
20			6	185515.4	39.99	156621.8	40.419	1	0	0	1
21			8	134977.5	40.623	134285.3	41.715	0	1	0	1
22		4	4	141495.7	44.341	137541.2	46.563	0	1	1	1
23			6	73480.6	46.644	73385.2	47.06	0	1	0	1
24			8	67165	47.38	66685	48.596	0	1	0	1
25		6	4	160185.5	51.07	159003.1	54.401	0	1	0	1
26			6	112433.7	54.867	103887.9	58.003	0	1	0	1
27			8	118921.9	53.88	115379	56.824	1	0	0	1
Total				74073.2	39.665	71894.66	41.181	2	25	3	24

Table 6 integrated the obtained results by 3 categories of number of surgeons, number of operation rooms, and number of patients. The table shows that RMGA produces better results than CGA in all cases. The table also shows that by increasing the number of patients, the average results and the average CPU times also increase. The number of operating rooms has a nonlinear relationship with the average results. In case there are 2 operating rooms, some patients may get treated late, due to a lack of resources. If the number of operating rooms is increased to 4, the number of operating rooms seems appropriate as the results are improved compared with the previous case. In the case of 6 operating rooms, although all patients are treated, more costs are imposed due to the unused operating rooms. By examining different levels of the number of surgeons, better solutions are obtained when the number of surgeons is increased. This will, however, increase the CPU time.

Table 6. Comparing the results of RMGA and CGA in each subset

		CGA		RMGA	
		Result	CPU time	Result	CPU time
Number of patients	20	31218.94	33.11	29308.22	33.95
	30	60059.72	39.63	57852.84	41.54
	40	130940.9	46.26	128522.9	48.06
Number of operation rooms	2	85621.36	35.25	83561.81	36.17
	4	59250.7	39.59	57823.4	41.15
	6	77347.53	44.16	74298.76	46.22
Number of surgeons	4	87480.88	37.89	86242.8	39.89
	6	71629.89	40.12	67595.26	41.35
	8	63108.82	40.99	61845.91	42.31

5-3- Comparison between RMGA and HGA

For more evaluation of the proposed algorithm, it is compared with a hybrid genetic algorithm, proposed to solve the nearest problem in the literature to current problem (Ala et al., 2020), namely HGA. They did not consider the surgeons in the scheduling procedure and only patients and operation rooms were considered in the scheduling procedure. We use the same test problem structured considered by (Ala et al., 2020). They considered 6 levels for the parameter of "Number of patients" and 3 levels for the parameter of "Number of operation rooms" (Number of patients = 10,20,30,40,50,100; Number of operation rooms = 5,10,15). By combination of these parameters 18 test problem is produced, and each problem is solved HGA and RMGA 1o times. For the parameter of operation time, uniform distribution U[1,99], and for the parameter of preparation time, uniform distribution U[1,9] are used. Regarding the considered parameters for HGA (Ala et al., 2020), the parameters of RMGA are set as follows: population size: 20; crossover rate :0.5; mutation rate: 0.2; α: 0.2; Max_itr: 200; n-good:5; n-bad: 5. Comparison results were illustrated in table 7.

Table 7. Comparing the results of RMGA with the result of the HGA

Problem number	Number of patients	Number of operation rooms	RMGA	HGA	RMGA	HGA	T	d.f	t	Sig
1	10	5	922.3	928	5.2	9.5	1.67	14	1.75	No
2	10	10	1189.7	1,202.50	6.8	12	2.92	14	1.75	Yes
3	10	15	1498.9	1,511.40	6.8	12.3	2.81	14	1.76	Yes
4	20	5	1164.2	1,171.30	3.8	7.5	3.8	28	1.77	Yes
5	20	10	1590	1,602.60	8	14.7	3.35	29	1.74	Yes
6	20	15	1902.5	1,917.00	11.4	22.5	2.56	28	1.73	Yes
7	30	5	1910.9	1,931.00	8	16	6.14	43	1.76	Yes
8	30	10	2265.9	2,279.40	11.5	22.9	2.9	43	1.75	Yes
9	30	15	2580.8	2,598.40	9.3	16.3	5.13	46	1.77	Yes
10	40	5	2693.1	2,703.70	13.2	24.5	2.41	60	1.73	Yes
11	40	10	3132.2	3,148.20	16.1	30.1	2.97	60	1.74	Yes
12	40	15	3453.1	3,492.70	14	24.5	8.88	62	1.74	Yes
13	50	5	2901.8	2,940.00	7.5	14.7	16.38	73	1.73	Yes
14	50	10	3207.2	3,219.00	9.2	16.9	4.35	76	1.74	Yes
15	50	15	3687.2	3,691.00	12.3	22.9	1.04	75	1.73	No
16	100	5	5526.5	5,594.00	18.2	34.2	17.43	151	1.73	Yes
17	100	10	5842.2	5,871.00	8.2	14.6	17.21	156	1.77	Yes
18	100	15	6417.8	6,442.10	13.8	26.1	8.22	151	1.75	Yes

In this section, the hypothesis test is used to determine the significance differences and, $\mu 1 - \mu 2 > 0$ was compared to $\mu 1 - \mu 2 = 0$ (null hypothesis). The difference of total final times follows the normal distribution, and the amount of α was 0.05. If the hypothesis is correct, the random variable T had a T-distribution. Formulas (27) and (28) define t-test and d.f (degree of freedom).

$$T = (\bar{X}1 - \bar{X}2)/\sqrt{(S_1^2/n_1) + (s_2^2/n_2)} \tag{27}$$

$$\text{d.f}= \left({S_1^2}/{n_1} + {s_2^2}/{n_2}\right)^2 / \left(\frac{\left({S_1^2}/{n1}\right)^2}{n_1-1} + \frac{\left({s_2^2}/{n2}\right)^2}{n_2-1} \right) \tag{28}$$

The critical value of C was extracted from prob(T>C) = α =0.05. For instance, in table 7, the sample size of data equals n1 = n2 =10, and the amount of average for RMGA and HGA are X1 = 922.3 and X2 = 928, respectively. As regards standard deviations, figures for the RMGA and HGA are S1 = 5.2 and S2 = 9.5, respectively. There is not significant difference among solutions, since T= 1.67 < t = 1.75. The superiority of RMGA was obvious compare to HGA due to significant differences in all except in two cases (Problem numbers 1 and 15).

5-4- Discussion

Two main differences between RMGA and a conventional GA are as follows:

_In the proposed algorithm the best and the worst solutions during the optimization process are saved in two sets of good and bad role models, respectively.

_ In RMGA, three chromosomes cooperates to perform the mutation operator: a chromosome is considered as a follower, and two chromosomes as influencer. one influencer is selected from good role model set and another is selected from bad role model set. Whereas in the standard genetic algorithm, the mutation operator requires only one randomly selected chromosome.

_ In the proposed algorithm the similarity procedure is used in the crossover operator.

the better performance of RMGA is shown in the comparisons. One of the main features of RMGA is its mutation procedure. The mutation operator in RMGA includes two steps. In the first Step, the random chromosome from the population (follower) obtains attributes of a member of good role model set (good influencer). The good influencer has better fitness value than other chromosomes in the population. It means that it has desirable attributes in its chromosome. The first step of the mutation operator tries to convey these desirable attributes to the follower chromosome. It is reasonable that we conclude that this step improves the fitness functions of the chromosomes.

On the other hand, in the second step of the mutation operator, the follower tries to remove some characteristics of a bad chromosome (bad influencer) from itself. A bad chromosome has worse fitness value than chromosomes of the population. It is because some undesirable attribute in its chromosome structure. Eliminating this bad attributes may cause an enhancement of the quality of the follower chromosome.

Although the first step of the mutation operator increase the similarity of chromosomes in the population, the second step of the mutation operator prevents it. In other words, the second step of mutation operator, enhances the diversity of chromosomes and delays the convergency of the genetic algorithm and decrease the probability of obtaining local optimum solutions.

6-Conclusion and future directions

In this study, the integrated operating room planning and scheduling problem was investigated at the operational level with aims to minimize the completion time of surgeries, minimize the surgeons' free time window, and minimize operating rooms' overtime and idle time costs, considering uncertainty in the surgery times, sequence-dependent operating rooms' preparation, and a short-term daily time horizon. The problem was solved by an algorithm inspired from the role model concept in sociology using simulating and differentiating procedures, namely RMGA. RMGA is compared with two other algorithms. The first compared algorithm is a conventional genetic algorithm namely CGA and the second one is a hybrid genetic algorithm proposed for the nearest problem in the literature (Ala et al., 2020), namely HGA. The comparison between RMGA and CGA shows that RMGA provides better results than CGA. In addition to the average solutions, the average solving times were also calculated for both algorithms. CGA offered a better average solving time than the proposed algorithm. It may be concluded from results, that the availability of resources played a key role in the patient scheduling process. Lack of access to the required resources during surgery could increase patients' waiting time and their dissatisfaction. Moreover, improper use of resources and improper planning could increase hospital costs due to unused resources or overworking. On the other hand, the comparison between RMGA and HGA shows the superiority of RMGA compare to HGA.

Considering three levels of decision-making in the patient scheduling process would be effective, Therefore, the combination of different levels of decision-making could lead to better decisions in terms of patient scheduling and resource allocation. In future studies, uncertainties such as resource unavailability and the arrival of emergency patients may be investigated in the proposed problem. Considering intensive care units and post-anesthesia care units may also be examined in the scheduling process.

References

Addis, B., Carello, G., Grosso, A., & Tànfani, E. (2016). Operating room scheduling and rescheduling: a rolling horizon approach. *Flexible Services and Manufacturing Journal*, *28*(1-2), 206-232.

Agnetis, A., Coppi, A., Corsini, M., Dellino, G., Meloni, C., & Pranzo, M. (2014). A decomposition approach for the combined master surgical schedule and surgical case assignment problems. *Health care management science*, *17*(1), 49-59.

Ahmadi-Javid, A., Jalali, Z., & Klassen, K. J. (2017). Outpatient appointment systems in healthcare: A review of optimization studies. *European Journal of Operational Research*, *258*(1), 3-34.

Akbarzadeh, B., Moslehi, G., Reisi-Nafchi, M., & Maenhout, B. (2020). A diving heuristic for planning and scheduling surgical cases in the operating room department with nurse re-rostering. *Journal of scheduling, 23*(2), 265-288.

Ala, A., Torkayesh, S. E., Torkayesh, A. E., & Iranizad, A. (2020). A hybrid genetic algorithm for appointment scheduling in a health examination system. *International Journal of Value Chain Management, 11*(4), 293-310.

Aringhieri, R., Landa, P., Soriano, P., Tànfani, E., & Testi, A. (2015). A two level metaheuristic for the operating room scheduling and assignment problem. *Computers & Operations Research, 54*, 21-34.

Arnaout, J.-P. (2010). Heuristics for the maximization of operating rooms utilization using simulation. *Simulation, 86*(8-9), 573-583.

Batun, S., Denton, B. T., Huschka, T. R., & Schaefer, A. J. (2011). Operating room pooling and parallel surgery processing under uncertainty. *INFORMS journal on Computing, 23*(2), 220-237.

Breuer, D. J., Lahrichi, N., Clark, D. E., & Benneyan, J. C. (2020). Robust combined operating room planning and personnel scheduling under uncertainty. *Operations Research for health care, 27*, 100276.

Cardoen, B., Demeulemeester, E., & Beliën, J. (2010). Operating room planning and scheduling: A literature review. *European Journal of Operational Research, 201*(3), 921-932.

Ciavotta, M., Dellino, G., Meloni, C., & Pranzo, M. (2010). *A rollout algorithmic approach for complex parallel machine scheduling in healthcare operations.* Paper presented at the XXXVI International ORAHS Conference.

Conforti, D., Guerriero, F., & Guido, R. (2010). *A multi-objective block scheduling model for the management of surgical operating rooms: New solution approaches via genetic algorithms.* Paper presented at the 2010 IEEE workshop on health care management (WHCM).

Denton, B. T., Miller, A. J., Balasubramanian, H. J., & Huschka, T. R. (2010). Optimal allocation of surgery blocks to operating rooms under uncertainty. *Operations research, 58*(4-part-1), 802-816.

Díaz-López, D., López-Valencia, N., González-Neira, E., Barrera, D., Suárez, D., Caro-Gutiérrez, M., & 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*(4), 409-422.

Dios, M., Molina-Pariente, J. M., Fernandez-Viagas, V., Andrade-Pineda, J. L., & Framinan, J. M. (2015). A decision support system for operating room scheduling. *Computers & Industrial Engineering, 88*, 430-443.

Doulabi, S. H. H., Rousseau, L.-M., & Pesant, G. (2014). *A constraint programming-based column generation approach for operating room planning and scheduling.* Paper presented at the International conference on ai and or techniques in constriant programming for combinatorial optimization problems.

Durán, G., Rey, P. A., & Wolff, P. (2017). Solving the operating room scheduling problem with prioritized lists of patients. *Annals of Operations Research, 258*(2), 395-414.

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.

Gartner, D., & Padman, R. (2017). Mathematical programming and heuristics for patient scheduling in hospitals: A survey. In *Handbook of research on healthcare administration and management* (pp. 627-645): IGI Global.

Ghazalbash, S., Sepehri, M. M., Shadpour, P., & Atighehchian, A. (2012). Operating room scheduling in teaching hospitals. *Advances in Operations Research, 2012*.

Guido, R., & Conforti, D. (2017). A hybrid genetic approach for solving an integrated multi-objective operating room planning and scheduling problem. *Computers & Operations Research, 87*, 270-282.

Hashemi Doulabi, S. H., Rousseau, L.-M., & Pesant, G. (2016). A constraint-programming-based branch-and-price-and-cut approach for operating room planning and scheduling. *INFORMS journal on Computing, 28*(3), 432-448.

Hulshof, P. J., Kortbeek, N., Boucherie, R. J., Hans, E. W., & Bakker, P. J. (2012). Taxonomic classification of planning decisions in health care: a structured review of the state of the art in OR/MS. *Health systems, 1*(2), 129-175.

Jebali, A., Alouane, A. B. H., & Ladet, P. (2006). Operating rooms scheduling. *International Journal of Production Economics, 99*(1-2), 52-62.

Kamran, M. A., Karimi, B., & Dellaert, N. (2020). A column-generation-heuristic-based benders' decomposition for solving adaptive allocation scheduling of patients in operating rooms. *Computers & Industrial Engineering, 148*, 106698.

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.

Latorre-Núñez, G., Lüer-Villagra, A., Marianov, V., Obreque, C., Ramis, F., & Neriz, L. (2016). Scheduling operating rooms with consideration of all resources, post anesthesia beds and emergency surgeries. *Computers & Industrial Engineering, 97*, 248-257.

Lin, Y.-K., & Chou, Y.-Y. (2020). A hybrid genetic algorithm for operating room scheduling. *Health care management science, 23*(2), 249-263.

Marques, I., & Captivo, M. E. (2015). Bicriteria elective surgery scheduling using an evolutionary algorithm. *Operations Research for health care, 7*, 14-26.

Marques, I., Captivo, M. E., & Pato, M. V. (2012). An integer programming approach to elective surgery scheduling. *OR spectrum, 34*(2), 407-427.

Marques, I., Captivo, M. E., & Pato, M. V. (2014). Scheduling elective surgeries in a Portuguese hospital using a genetic heuristic. *Operations Research for health care, 3*(2), 59-72.

Marques, I., Captivo, M. E., & Pato, M. V. (2015). A bicriteria heuristic for an elective surgery scheduling problem. *Health care management science, 18*(3), 251-266.

Mateus, C., Marques, I., & Captivo, M. E. (2018). Local search heuristics for a surgical case assignment problem. *Operations Research for health care, 17*, 71-81.

Merton, R. K. (1957). The Role-Set: Problems in Sociological Theory. *The British Journal of Sociology, 8*, 106-120.

Molina-Pariente, J. M., Hans, E. W., & Framinan, J. M. (2018). A stochastic approach for solving the operating room scheduling problem. *Flexible services and manufacturing journal, 30*(1-2), 224-251.

Molina, P., Jose M, Fernandez-Viagas, V., & Framinan, J. M. (2015). Integrated operating room planning and scheduling problem with assistant surgeon dependent surgery durations. *Computers & Industrial Engineering, 82*, 8-20.

Molina, P., Jose M, Hans, E. W., & Framinan, J. M. (2018). A stochastic approach for solving the operating room scheduling problem. *Flexible Services and Manufacturing Journal, 30*(1-2), 224-251.

Moosavi, A., & Ebrahimnejad, S. (2018). Scheduling of elective patients considering upstream and downstream units and emergency demand using robust optimization. *Computers & Industrial Engineering, 120*, 216-233.

Pham, D.-N., & Klinkert, A. (2008). Surgical case scheduling as a generalized job shop scheduling problem. *European Journal of Operational Research, 185*(3), 1011-1025.

Riise, A., & Burke, E. K. (2011). Local search for the surgery admission planning problem. *Journal of Heuristics, 17*(4), 389-414.

Riise, A., Mannino, C., & Burke, E. K. (2016). Modelling and solving generalised operational surgery scheduling problems. *Computers & Operations Research, 66*, 1-11.

Roland, B., Di Martinelly, C., & Riane, F. (2006). *Operating theatre optimization: A resource-constrained based solving approach.* Paper presented at the 2006 International conference on service systems and service management.

Roland, B., Di Martinelly, C., Riane, F., & Pochet, Y. (2010). Scheduling an operating theatre under human resource constraints. *Computers & Industrial Engineering, 58*(2), 212-220.

Roshanaei, V., Booth, K. E., Aleman, D. M., Urbach, D. R., & Beck, J. C. (2020). Branch-and-check methods for multi-level operating room planning and scheduling. *International Journal of Production Economics, 220*, 107433.

Saadouli, H., Jerbi, B., Dammak, A., Masmoudi, L., & Bouaziz, A. (2015). A stochastic optimization and simulation approach for scheduling operating rooms and recovery beds in an orthopedic surgery department. *Computers & Industrial Engineering, 80*, 72-79.

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.

Tan, Y., El Mekkawy, T., Peng, Q., & Oppenheimer, L. (2007). Mathematical programming for the scheduling of elective patients in the operating room department. *Proceedings of the Canadian Engineering Education Association (CEEA)*.

Testi, A., & Tànfani, E. (2009). Tactical and operational decisions for operating room planning: Efficiency and welfare implications. *Health care management science, 12*(4), 363.

Testi, A., Tanfani, E., & Torre, G. (2007). A three-phase approach for operating theatre schedules. *Health care management science, 10*(2), 163-172.

Van Huele, C., & Vanhoucke, M. (2014). Analysis of the integration of the physician rostering problem and the surgery scheduling problem. *Journal of medical systems, 38*(6), 43.

Van Oostrum, J. M., Bredenhoff, E., & Hans, E. W. (2010). Suitability and managerial implications of a master surgical scheduling approach. *Annals of Operations Research, 178*(1), 91-104.

Vijayakumar, B., Parikh, P. J., Scott, R., Barnes, A., & Gallimore, J. (2013). A dual bin-packing approach to scheduling surgical cases at a publicly-funded hospital. *European Journal of Operational Research, 224*(3), 583-591.

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.

Zhao, Z., & Li, X. (2014). Scheduling elective surgeries with sequence-dependent setup times to multiple operating rooms using constraint programming. *Operations Research for health care, 3*(3), 160-167.

Zhu, S., Fan, W., Liu, T., Yang, S., & Pardalos, P. M. (2020). Dynamic three-stage operating room scheduling considering patient waiting time and surgical overtime costs. *Journal of Combinatorial Optimization, 39*(1), 185-215.

Zhu, S., Fan, W., Yang, S., Pei, J., & Pardalos, P. M. (2019). Operating room planning and surgical case scheduling: a review of literature. *Journal of Combinatorial Optimization, 37*(3), 757-805.