Medical & Biological Engineering & Computing (2022) 60:1295–1311
https://doi.org/10.1007/s11517-022-02513-3

ORIGINAL ARTICLE

Challenges and solutions for the integrated recovery room planning and scheduling problem during COVID-19 pandemic

Marouene Chaieb[1,2] · Dhekra Ben Sassi[1,3] · Jaber Jemai[4] · Khaled Mellouli[2]

Received: 17 April 2021 / Accepted: 10 January 2022 / Published online: 22 March 2022

Abstract

This study presents an efficient solution for the integrated recovery room planning and scheduling problem (IRRPSP). The complexity of the IRRPSP is caused by several sources. The problem combines the assignment of patients to recovery rooms and the scheduling of caregivers over a short-term planning horizon. Moreover, a solution of the IRRPSP should respect a set of hard and soft constraints while solving the main problem such as the maximum capacity of recovery rooms, the maximum daily load of caregivers, the treatment deadlines, etc. Thus, the need for an automated tool to support the decision-makers in handling the planning and scheduling tasks arises. In this paper, we present an exhaustive description of the epidemiological situation within the Kingdom of Saudi Arabia, especially in Jeddah Governorate. We will highlight the importance of implementing a formal and systematic approach in dealing with the scheduling of recovery rooms during extreme emergency periods like the COVID-19 era. To do so, we developed a mathematical programming model to present the IRRPSP in a formal way which will help in analyzing the problem and lately use its solution for comparison and evaluation of our proposed approach. Due to the NP-hard nature of the IRRPSP, we propose a hybrid three-level approach. This study uses real data instances received from the Department of Respiratory and Chest Diseases of the King Abdulaziz Hospital. The computational results show that our solution significantly outperforms the results obtained by CPLEX software with more than 1.33% of satisfied patients on B1 benchmark in much lesser computation time (36.27 to 1546.79 s). Moreover, our proposed approach can properly balance the available nurses and the patient perspectives.

Keywords Patient assignment · COVID-19 pandemic · Combinatorial optimization problem · Decision support system · Healthcare management

1 Introduction

The COVID-19 pandemic is an extremely dangerous ongoing global pandemic that was first identified in Wuhan, China, in December 2019 [18]. On 17 April 2021, more than 140 million cases of coronavirus have been reported in more than 200 countries and territories around the world, resulting in more than 3 million deaths [19]. This global health crisis can be considered the greatest challenge that humanity has faced from the Second World War which results in an unprecedented socio-economic crisis. It has the potential to cause devastating economic, social, education, and political disruptions that eventually affect negatively the countries it touches. It has led to the closing of schools, colleges, and universities in more than 200 countries affecting approximately 98.5% of the students around the world. In this paper, we emphasize on the situation in the Kingdom of Saudi Arabia (KSA) which is one of the most affected countries by the COVID-19 pandemic, especially in the high number of daily infections. As of 14 April 2021, the kingdom has more than 400,000 confirmed cases, the highest among the Arabian Gulf States with more than 6800 deaths [19]. This huge number of confirmed cases pushes KSA to follow a set of painful precautions such as declaring a comprehensive quarantine for a long period, starting from September 2020.

To handle extreme emergency cases during crisis periods like the COVID-19 pandemic, healthcare institutions must be equipped with all necessary tools and resources to

✉ Marouene Chaieb
m.chaieb@psau.edu.sa; chaieb.marouene@live.fr

Extended author information available on the last page of the article.

expedite the healing and recovery of different types of patients. COVID-19 patients show mainly respiratory symptoms that need urgent treatments in specialized rooms by highly skilled caregivers. Such resources namely special rooms and skilled personnel may not be available at peak periods and may be also overwhelmed by the flow of patients. To solve such a puzzle of planning and scheduling recovery rooms under inherently complex operational constraints, an efficient approach is needed.

In this paper, we present and develop an IRRPSP solving strategy based on the decomposition of the problem into three levels. Our main challenge is to find the best assignment of patients to rooms and scheduling of caregivers to those patients needing intensive care. Such solution should be obtained considering a low availability of healthcare givers, the lack of resources (number of medial beds, number of respirators, etc.), a quick overburdening of hospitals during the COVID-19 crisis since the demand of care exceeds the capacity of hospitals, and the increase of infections during an ongoing pandemic which may result in being highly contagious to both staff and other patients.

Our main contribution is to find a creative solving approach to the integrated recovery room planning and scheduling problem which to the best of our knowledge is not well studied in the literature with its two components (assignment of patients to recovery rooms and assignment of nurses to patients). A study that aims to find a creative solution for building a more resilient, sustainable, and inclusive scheduling of patients and nurses in a crisis context, although the new interesting point revealed by this work is the resulting performance against the CPLEX results since the computational analysis demonstrated the capability of the suggested approach to deal efficiently with the trade-off between the available nurses and patient perspectives in reasonable computational times.

The IRRPSP can be modeled as a well-known combinatorial optimization problem (COP) which is the patient assignment and scheduling to operatoring rooms (PASOR) in hospitals that involves optimizing simultaneously different objectives such as maximizing the number of assigned patients to rooms and maximizing the number of satisfied tasks by caregivers. To model the IRRPSP, we developed an integer linear mathematical programming model that aims to maximize the utilization of the COVID-19 department while satisfying some constraints.The IRRPSP is an NP-hard problem that cannot be solved optimally using traditional solving methods. In order to efficiently solve the problem, a three-level hybrid algorithm is proposed that will be tested on a set of real-life data instances. A comparative study will be conducted to demonstrate the performance of the modeling and solving process in terms of the overall quality of the provided solution compared to an exact solution using a CPLEX solver.

The remainder of this paper is organized as follows. The background of the tackled problem and the related literature will be detailed in Section 2. Section 3 will present a detailed description of the problem and a mathematical formulation will be provided. Section 4 will be devoted to detail the solution of the three-level hybrid algorithm. The performed simulations, experimental results, and validation tests will be detailed in Section 5. By the end, we present our conclusions and the future research perspectives.

2 Literature review

The related literature to the IRRPSP will be reviewed in this section. For each paper, we will discuss the tackled problem, the modeling approach, the solving approach, the data instances used to prove the efficiency of the proposed approach (real-life data instances, simulated data instances), and the obtained results. The literature on the patient assignment and scheduling problem is discussed in Section 2.1. The findings of the PASOR literature and the main contributions of the paper will be detailed in the second subsection.

2.1 Literature review of the patient assignment and scheduling

There exists a large set of scientific researches in the literature that study the PASOR which is similar to our studied problem (IRRPSP) [17]. The considered problem has been widely tackled, and numerous recent related works are available. Here, we only discuss a set of studies that are most related to our problem and solution methodology.

In 2020, [7] addressed the shift scheduling problem of physicians during the COVID-19 pandemic. In their study, the authors developed a decision support system (DSS) that helps hospitals in generating balanced shifts for both regular and COVID-19 workloads. To model the shift scheduling problem, the authors presented a mixed integer programming which was subsequently solved using Gurobi software. According to the authors, the proposed DSS is able to generate schedules for physicians in a very short time. The scheduling of the staff is distributed evenly and the duration of exposure to the virus is reduced. However, we can remark that the main limitation of this work is to not solve the problem using a powerful algorithm (heuristic or a metaheuristic algorithm) since it will not be possible to optimally solve the problem if the number of patients blow up and the problem becomes complicated.

In 2016, [9] studied the PASOR by combining the assignment of surgeries to operating rooms and scheduling over a short-term planning horizon. To model the studied problem, the authors developed a mathematical programming model. To solve the proposed model, the authors developed a branch-and-price-and-cut algorithm based on a constraint programming model. In addition, to improve the efficiency of the constraint programming model, [9] developed a set of dominance rules and a fast infeasibility-detection solution using a multidimensional knapsack problem. According to the authors, the computational results on generated (artificial) instances indicate that their approach has an average optimality gap of 2.81% and significantly outperforms a compact mathematical formulation used in similar research. However, the author solved the model using only a branch-and-price-and-cut algorithm which we think is not enough to prove the efficiency of their approach. Thus, using more powerful algorithms is needed to validate the obtained results.

In 2020, [2] dealt with the problem of urgent surgeries assignment in an existing and in-course operating schedule by assigning nonelective patients. The aim was the assignment of the maximal amount of nonelective cases and global operating block overtime cost-efficiency. The methodology of the paper was based on dividing patients into three classes which are emergent, urgent, and work in case. To prove the efficiency of the proposed approach, a set of experiments were conducted on 5 generated data instances and results show an improvement on both operating rooms capacity filling and supplementary hours overrun limitation. However, the main concern with their study is the small volume of operation theater schedules used to prove the efficiency of the modeling and solving approach.

In 2020, [1] addressed the allocation of resources in the operating room department. The main aim was to maximize the operating room profit using the operating room department as efficiently as possible. To do so, the authors suggested to integrating three main components which are (i) the surgical case planning and scheduling problem, (ii) the nurse re-rostering decision, and (iii) the nurse assignment to specific patients. To model the problem, the authors presented a mathematical formulation that was solved by a two-phase heuristic that uses the linear programming solution generated via column generation. The proposed modeling and solving approaches were tested on two types of datasets: (i) the real-life data received from the Sina Iranian Hospital and (ii) the artificial datasets generated in a controlled and structured manner. On the other hand, the choice of the column generation-based diving heuristic solution was not argued by the authors which may present a concern regarding the validity of the results.

In 2019, [14] treated a weekly operation scheduling problem of elective surgery according to the block scheduling policy. According to the authors, two main objectives to reach by solving the considered problem which are balancing the overtime and decreasing the unused time of each surgeon's blocks. To model the problem, the authors choose to decompose it into two sub-problems. The first sub-problem aims to allocate patients to blocks and achieve the balance of the blocks to the same surgeon. In the second sub-problem, three parameters are defined which are the operating room, the operation date, and the operation sequence. Both subproblems were modeled using integer programming models and solved using CPLEX optimizer. According to [14], their proposed approach can present a support decision tool to balance surgeons' workload among blocks and can improve patient satisfaction. In addition, the computational experiments are conducted on generated artificial data instances illustrating their applicability to the problem in the operating theater.

In 2019, [15] dealt with the surgery assignment problem to determine the number and type of surgeries to schedule in a 2-week planning horizon where each operating session is assigned to a surgical specialty according to a fixed grid. The main aim is to maximize the expected operating theaters throughput. To do so, the authors modeled the considered problem using a stochastic model, which was solved using a sample average approximation technique that uses a limited number of scenarios. To prove the efficiency of the proposed approach, the authors executed a set of tests on a real case study with real data from a leading European children's hospital. The computational study shows that it is fundamental to consider the stochastic nature of the problem for both the lengths of stay and the surgery times. However, the authors developed a comparative study in which they compared their stochastic with the deterministic model which can be ambiguous since they are considered two different models; thus, this comparison will be inconsistent.

In 2015, [16] discussed the problem of scheduling elective surgery patients in the orthopedic surgery division of a Tunisian hospital considering two types of resources, which are the recovery rooms and the operating beds. The authors aim to optimize the assignment of surgeries to operating rooms and the planning of the recoveries. To do so, they divided the modeling and the solving of the main problem into two phases. In the first stage, they developed a mixed-integer program to model the problem as a knapsack problem with the aim of choosing the operations to be scheduled on the selected day. In a second stage, to model a discrete event simulation, the authors developed a bi-objective mixed-integer program to compare the head surgeon's actual practice and the new model. According to the authors, their proposed approach indicates that a substantial amount of operations could be saved if the method is implemented.

In 2019, [8] addressed the scheduling problem of inpatient surgeries to improve the impact on the quality and safety of surgery. The authors aim to improve the compatibility level within the surgical teams by incorporating the decision-making styles of the surgical team members (in an operating room scheduling problem). To model the problem, the authors proposed a comprehensive multi-objective programming model that aims to ensure three objectives, which are the minimization of the total cost, the maximization of the patients' satisfaction, and the maximization of the aggregation of compatibilities of the surgical teams. The authors developed two metaheuristics which are the NSGA-II and the MOPSO to find Pareto solutions. Furthermore, the PROMETHEE-II method was used to select the best among the obtained Pareto solutions. According to the authors, the results generated by MOPSO indicate better performance compared to NSGA-II. It is important to mention here that the resulting schedule is limited since it is a short-horizon schedule (daily). Thus, it is inconsistent to validate the proposed approach using a daily schedule. It requires extending the approach to be able to define a weekly or semi-monthly schedule that can be used for evaluating the approach efficiency.

In 2019, [4] dealt with the operating room planning problem while considering uncertainty in surgery durations is incorporated in this problem. The main aim of the paper is to allow sicker patients to have early access to surgery. To model the studied problem, the authors presented a stochastic mixed-integer mathematical formulation that aims to improve the assignment of surgeries and to minimize the total overtime that exceeds the available time durations for surgeries. To solve the proposed model, the authors suggested three solving methods, which are (i) a Tabu Search metaheuristic (TS), (ii) a fastest ascent local search, and (iii) the sample average approximation method. The experimental study shows that the TS metaheuristic provides effective solutions within reasonable computation times. This interpretation was not justified by the authors to detail the main reasons that make the TS suitable to solve the studied problem than other solving techniques.

2.2 Findings of the literature review and the contribution of the paper

There exist numerous papers in the literature that treat the planning and scheduling of patients in the operating rooms department. The topic has been extensively investigated which highlights its importance to the patients and hospitals by realizing both health and cost-efficiency objectives. In this context, the experiments of each paper show an improvement on both operating rooms capacity filling and supplementary hours overruns limitation. Multiple decision support tools have been developed to assist managers in taking better and optimal assignment and scheduling decisions. In addition, multiple works in the literature show that developing decision tools presented strategies in developed countries' hospitals could help them limit their healthcare budget overruns. On the other hand, the contribution of this paper is twofold. First, we consider the patient assignment and scheduling problem during the COVID-19 pandemic, which is, to the best of our knowledge, not yet addressed in the literature, and this is the first attempt to solve it. Second, we present a three-level hybrid algorithm to solve the integrated recovery room planning and scheduling problem. By the end, the experimental study shows that the suggested methodology realizes (near-)optimal solutions and outperforms the CPLEX software solution.

3 Problem description and formulation

The planning and scheduling of recovery rooms during the COVID-19 epidemic is a difficult process. Its real difficulty came from the huge number of patients needing treatment. Due to the short period of arrival, the operating room schedule has to be optimized. Thus, the need for a decision support system, helping the decision-makers in the hospital to handle this big flow of patients, is not a choice but an obligation.

In this section, we go over detailing the main objective to reach after solving the studied problem by giving a detailed description. Then, we will present a problem formulation by developing a mathematical model.

3.1 Problem description

The studied problem aims essentially to optimize two goals which are (i) the assignment of patients to recovery rooms and (ii) the scheduling of nurses to patients.

According to the Saudi Ministry of Health, four categories of COVID-19 confirmed patients are considered. The first type is the asymptomatic cases which do not requires any treatment and has to follow instructions and recommendations published by the Saudi Ministry of Health. The second category is composed of cases from middle to moderate state which requires to treat symptoms and having to follow instructions and recommendations published by the Saudi Ministry of Health. The third category consists of severe cases that must be admitted to the hospital and treated by caregivers in the Intensive Care Unit. The last category contains the critical case patients who are also called to be admitted to the hospital with more intensive care. In the following table, we give more details about each category. In our context, we will consider only the third and the fourth categories since they must be admitted to the hospitals. At a daily meeting of the medical staff, the state of patients is discussed to decide and many scenarios are possible. They start by checking if the patient

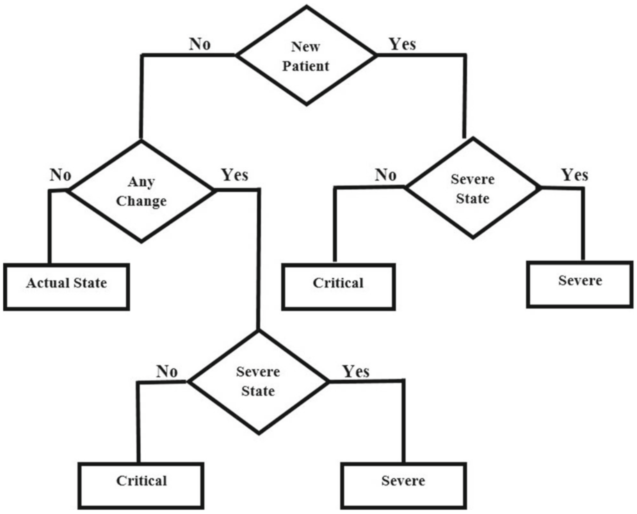

Fig. 1 Patient categories based on the state

is new or not. If he is a previous patient, then they check if there is "any change" from his previous state; if yes, they will do another check to precise if they have to consider him as a critical or a severe patient. On the other hand, if the patient is new, the medical staff will directly do the second check. The following flowchart (see Fig. 1) summarizes all possible states.

The problem of the daily patient assignment to recovery rooms scheduling of the COVID-19 division Jeddah hospital can be summarized as follows: a set of p patients to be scheduled in k recovery room given an available rooms' capacity c. Two types of resources are considered: recovery beds (RB) and respiratory machines (RM). These types of resources are used mainly by two types of recovery room schedules which are considered in the hospitals. The first type treats the severe patients without considering the critical cases; thus, rooms contain only recovery beds and are designed in our work as Recovery Beds Rooms (RBR). The other one treats the critical cases using special respiratory machines and this type of room is designed in our work as Recovery Respiratory Rooms (RRR). After assigning patients to rooms, the set of caregivers must be scheduled based on the type of required care. In this respect, only skilled caregivers may serve patients of RRR, which are NRRR, and other nurses can satisfy tasks of patients in RBR, which are NRBR.

3.2 Problem formulation

In this subsection, we will provide a mathematical model for the assignment and scheduling of patients to the recovery rooms which is a way for good research modeling since it simplifies the essential concepts and the relationships between them. Besides, the provided mathematical model facilitates and aids for clear thinking and provides an explicit connection to the existing body of knowledge by studying the effects of different components, and to make predictions about its behavior. In the following subsection, we will detail the set of notations, the set of decision variables to use, the objective functions to accomplish, and the list of constraints to respect.

3.2.1 Notations

In the following Table 1, we mainly present the required notations for developing our mathematical model which are the sets, the deterministic parameters, and the decision variables.

3.3 Mathematical model

In this subsection, we propose a mathematical formulation of the objective function and different constraints of the considered problem.

3.3.1 Objective function

We recall here that the main aim of solving the integrated recovery room planning and scheduling problem is to maximize essentially two factors which are:

- The number of assigned patients to rooms.
- The number of treated patients by nurses.

So to achieve both previous objectives, the model must contain the following objective function:

$$Max \sum_{i \in S} \sum_{k \in R} \sum_{a \in G} x_{ik} + y_{ia} \forall i \in S, \forall k \in R, \forall a \in G \tag{1}$$

3.3.2 Constraints

The IRRPSP is considered a highly constrained problem with a big number of both soft and hard constraints. In the following listing are of constraints to be respected.

To ensure that a severe state patient is either assigned to a room with recovery beds or not assigned at all, we have to add the following constraint to the model:

$$\sum_{i \in S} \sum_{l \in B} x_{il} \leq 1 \forall i \in S, \forall l \in B \tag{2}$$

To ensure that a critical patient is either assigned to a room with respiratory machines or not assigned at all, we have to add the following constraint to the model:

$$\sum_{j \in C} \sum_{k \in R} x_{jk} \leq 1 \forall j \in C, \forall k \in R \tag{3}$$

Table 1 Problem notations

Sets	Description
S:	The set of patients with severe state.
C:	The set of patients with critical state.
B:	The set of rooms with recovery beds.
R:	The set of rooms with respiratory machines.
G:	The set of nurses able to treat patients with severe state.
N:	The set of qualified nurses to treat patients with critical state.
Deterministic parameters	Description
$i = 0, 1, 2, ..., n$:	The index of patients with severe state.
$j = 0, 1, 2, ..., m$:	The index of patients with critical state.
$l = 0, 1, 2, ..., p$:	The index of room with recovery beds.
$k = 0, 1, 2, ..., o$:	The index of room with respiratory machines.
a	The index of nurses able to treat patients with severe state.
b	The index of qualified nurses to treat patients with critical state.
C_s	Capacity of the rooms with recovery beds.
C_c	Capacity of the rooms with respiratory machines.
e_i	The stating time of the treatment of the patient i.
l_i	The ending time of the treatment of the patient i.
S_i	The service time of the patient i.
g_a	The stating time of the duty of the nurse a.
h_a	The ending time of the duty of the nurse a.
Decision variables	Description
x_{ik} :	1 if the patient i is assigned to the room k 0, otherwise
y_{ia} :	1, if the nurse a is assigned to treat the patient i 0, otherwise.

To ensure that a severe state patient cannot be assigned to a room with respiratory machines, we have to add the following constraint to the model:

$$\sum_{i \in S} \sum_{k \in R} x_{ik} = 0 \forall i \in S, \forall k \in R \tag{4}$$

To ensure that a critical state patient cannot be assigned to a room with recovery beds, we have to add the following constraint to the model:

$$\sum_{j \in C} \sum_{l \in B} x_{jl} = 0 \forall j \in C, \forall l \in B \tag{5}$$

To ensure that a severe state patient cannot be assigned to a room with recovery beds, we have to add the following constraint to the model:

$$\sum_{j \in C} \sum_{l \in B} x_{jl} = 0 \forall j \in C, \forall l \in B \tag{6}$$

To ensure that severe state patients is either covered by a nurse able to treat patients with severe state or not handled at all, we have to add the following constraint to the model:

$$\sum_{i \in S} \sum_{a \in G} y_{ia} \leq 1 \forall i \in S, \forall a \in G \tag{7}$$

To ensure that critical state patients is either covered by a nurse able to treat patients with critical state or not handled at all, we have to add the following constraint to the model:

$$\sum_{j \in C} \sum_{b \in N} y_{jb} \leq 1 \forall j \in C, \forall b \in N \tag{8}$$

To ensure that severe state patients is not covered by a nurse allowed to treat only patients with critical state, we have to add the following constraint to the model:

$$\sum_{i \in S} \sum_{b \in N} y_{ib} = 0 \forall i \in S, \forall b \in N \tag{9}$$

To ensure that critical state patients is not covered by a nurse allowed to treat only patients with severe state, we have to add the following constraint to the model:

$$\sum_{j \in C} \sum_{a \in G} y_{ja} = 0 \forall j \in C, \forall a \in G \tag{10}$$

To ensure that all patients are treated within their time window, we have to add the following constraint to the model:

$$e_i \sum_{i \in S} \sum_{l \in B} x_{il} \leq S_i \leq l_i \sum_{i \in S} \sum_{l \in B} x_{il} \forall i \in S, \forall l \in B \tag{11}$$

To ensure that nurses are only assigned tasks within their working hours, we have to add the following constraint to the model:

$$g_a \sum_{i \in S} \sum_{l \in B} x_{il} \leq S_i \leq h_i \sum_{i \in S} \sum_{l \in B} x_{il} \forall i \in S, \forall l \in B \tag{12}$$

To ensure that the number of assigned severe state patients to a room with recovery beds does not exceed its capacity, we have to add the following constraint to the model:

$$\sum_{i=1}^{S} x_{il} \leq C_s \forall l \in B \tag{13}$$

To ensure that the number of assigned critical state patients to a room with respiratory machines does not exceed its capacity, we have to add the following constraint to the model:

$$\sum_{j=1}^{C} x_{jk} \leq C_c \forall k \in R \tag{14}$$

Now, we can present the final version of our mathematical model:

$$\sum_{i \in S} \sum_{k \in R} \sum_{a \in G} x_{ik} + y_{ia} \forall i \in S, \forall k \in R, \forall a \in G \tag{15}$$

$$\sum_{i \in S} \sum_{l \in B} x_{il} \leq 1 \forall i \in S, \forall l \in B \tag{16}$$

$$\sum_{j \in C} \sum_{k \in R} x_{jk} \leq 1 \forall j \in C, \forall k \in R \tag{17}$$

$$\sum_{i \in S} \sum_{k \in R} x_{ik} = 0 \forall i \in S, \forall k \in R \tag{18}$$

$$\sum_{j \in C} \sum_{l \in B} x_{jl} = 0 \forall j \in C, \forall l \in B \tag{19}$$

$$\sum_{j \in C} \sum_{l \in B} x_{jl} = 0 \forall j \in C, \forall l \in B \tag{20}$$

$$\sum_{i \in S} \sum_{a \in G} y_{ia} \leq 1 \forall i \in S, \forall a \in G \tag{21}$$

$$\sum_{j \in C} \sum_{b \in N} y_{jb} \leq 1 \forall j \in C, \forall b \in N \tag{22}$$

$$\sum_{i \in S} \sum_{b \in N} y_{ib} = 0 \forall i \in S, \forall b \in N \tag{23}$$

$$\sum_{j \in C} \sum_{a \in G} y_{ja} = 0 \forall j \in C, \forall a \in G \tag{24}$$

$$e_i \sum_{i \in S} \sum_{l \in B} x_{il} \leq S_i \leq l_i \sum_{i \in S} \sum_{l \in B} x_{il} \forall i \in S, \forall l \in B \tag{25}$$

$$g_a \sum_{i \in S} \sum_{l \in B} x_{il} \leq S_i \leq h_i \sum_{i \in S} \sum_{l \in B} x_{il} \forall i \in S, \forall l \in B \tag{26}$$

$$\sum_{i=1}^{S} x_{il} \leq C_s \forall l \in B \tag{27}$$

$$\sum_{j=1}^{C} x_{jk} \leq C_c \forall k \in R \tag{28}$$

4 Solving approach

The problem of patients' assignment and scheduling to recovery rooms addressed in this paper is really difficult due to its combinatorial nature. In addition, by the analysis of the number and type of constraints involved, which are sometimes conflicted, we conclude that the studied problem is an NP-hard problem. Thus, exact algorithms for solving large-sized instances of the problem are therefore unlikely. In this regard, we present in this section a three-level hybrid algorithm to solve the IRRPSP. Both first and second levels consist of defining an initial deterministic schedule, then an adaptation step is invoked to rectify the preschedule and improve the final solution. Figure 2 provides an overview of these phases of the solution procedure which will be described in the next three subsections.

4.1 Assignment of patients to recovery rooms

First, we aim to solve the assignment of patients to rooms depending on their states which will define the number of satisfied patients. To do so, we choose to first define a starting strategy where the initial solution is generated using the basic sweeping algorithm. The initial deterministic solution will provide a preschedule of assigned patients to the available recovery rooms and may improve the quality of the final solution.

An adaptation procedure is then executed to adapt the preschedule assignment to reach a feasible and robust

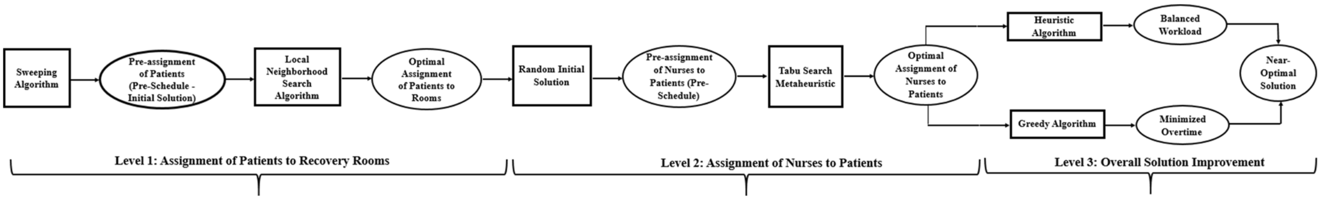
Sweeping Algorithm
Pre-assignment of Patients (Pre-Schedule - Initial Solution)
Local Neighborhood Search Algorithm
Optimal Assignment of Patients to Rooms
Random Initial Solution
Pre-assignment of Nurses to Patients (Pre-Schedule)
Tabu Search Metaheuristic
Optimal Assignment of Nurses to Patients
Heuristic Algorithm
Balanced Workload
Greedy Algorithm
Minimized Overtime
Near-Optimal Solution
Level 1: Assignment of Patients to Recovery Rooms
Level 2: Assignment of Nurses to Patients
Level 3: Overall Solution Improvement

Fig. 2 The three-level hybrid algorithm

solution with respect to the set of related soft and hard constraints and to maximize the occupation of the recovery rooms. This adaptation is done using a local neighborhood search algorithm which consists of essentially achieving two goals: (i) first search to regain the feasibility of the initial solution with respect to a set of constraints and then seeks to enhance its accuracy while maintaining its feasibility.

This number of satisfied patients (the output of the first component) will present the input of the Tabu solving algorithm to assign the required nurses depending on the type of recovery rooms

4.2 Assignment of nurses to patients

Second, we aim to solve the assignment of nurses to the set of patients depending on the type of recovery rooms (with respiratory machines or with recovery beds).

To do this, we start by generating a random initial solution by randomly combining the set of assignments of nurses to recovery rooms. Then, an adaptation procedure is invoked to rectify the preschedule of nurses to patients with respect to the room type. This adaption is executed using a Tabu Search metaheuristic which was in charge of essentially enhancing the actual assignments while preserving feasibility. TS metaheuristic is an evolved optimization problem-solving metaheuristic and was first proposed by Glover in [6]. TS is a metaheuristic that can guide the search process to ameliorate the neighborhood generation and evaluation mechanism [12]. The TS starts by defining an initial solution that can be feasible or unfeasible and it will be used to apply a set of neighborhood generation operators to move to another solution in the search space. In our case, we generate a random initial solution by randomly combining the set of visits. In order to explore the search space, the TS implements the concept of neighborhood generation operators which consists of doing some transformations on an input solution to create a set of other solutions named neighborhood.

Once an operator is applied and the neighborhood is generated, the algorithm will select the best newfound solution and consider it a new initial solution to continue exploring the search space. This basic search phase is continued until a kind of stability criterion is satisfied, more precisely until the number of iterations without improving reaches a predefined value. In this case, we use two different concepts to explore the search space which are:

- *The intensification* which is an intermediate-term memory search that consists of extensively searching around good solutions by using suitable operators and increasing the Tabu list size.
- *The diversification* which is an intermediate-term memory, searches that aim to explore new regions in the search space where good solutions can be found. It can be implemented using a dynamic Tabu list and some neighborhood operators.

Table 2 Characteristics of the benchmark set B1

ID	NCP	NSP	nRRR	nRBR	NRM	NRB	NRRR	NRBR
1	177	354	13	25	182	350	10	20
2	204	408	13	25	182	350	10	20
3	229	458	15	29	210	406	10	20
4	240	479	15	29	210	406	10	20

In both concepts, we mainly used three operators which are:

- *The Swap Operator*: that consists of taking two patients and swaps their position in the current solution.
- *The 2-opt Operator*: that takes as input two schedules and two different patients each per schedule. Then, it breaks the first schedule in its corresponding first point and completes it with the second part of the second schedule starting from point 2. The same action is done to complete the second schedule that will be completed by the second part of the first schedule.
- *Ejection Chains* that consists of taking two schedules defined by a starting and ending visits from the current solution. The considered chain will be copied from the current schedule to the next schedule iteratively until reaching the last schedule. Once an operator is applied and the neighborhood is generated, the algorithm will select the best newfound solution and consider it as a new initial solution to continue exploring the search space.

4.3 Overall solution improvement

The third step is considered an improvement phase which aims to minimize the set of unsatisfied tasks by essentially two ways:

- *Using the available overtime:* by violating the nurses time windows using the available overtime feature while maintaining the reliability of previous steps schedules. The main idea consists of defining the list of already assigned patients and then executing some greedy procedures to assign the available overtime.

Table 3 Characteristics of the benchmark set B2

ID	NCP	NSP	nRRR	nRBR	NRM	NRB	NRRR	NRBR
5	267	533	19	38	266	532	10	20
6	340	679	22	44	312	622	10	20
7	415	829	26	51	358	716	10	20
8	479	958	30	60	420	839	10	20

Table 4 Comparison of solution quality between CPLEX and our approach on benchmark B1

Id	NCP	NSP	CPLEX	Our approach	Gap %
1	177	354	1054	1054	0 %
2	204	408	1064	1064	0 %
3	229	458	1216	1232	1.32 %
4	240	479	1216	1232	1.32 %
Average					**0.66%**

- *Maximizing the recovery room utilization:* by balancing the workload between nurses in different recovery rooms profiting from the unused under-time in each recovery room, i.e., a nurse that has unused time and without more patients to cover can be assigned to another room with the same required skills.

Both solutions maximize the set of uncovered patients which means an improvement of the cost-efficiency.

5 Computational experiments and evaluation

In this section, the computational performance of the proposed methodology is reported and evaluated by solving 8 data instances containing more than 2300 patients with different optimization criteria and skill management policies. The considered measures are the optimality of the solution and the computational time needed to reach it. The main aim of the computational experiments is to assess the advantages of the proposed modeling and solving approach by extracting the main features that outperform similar approaches.

To do that, we first describe in Section 5.1 the computational environment and the benchmark instances. Then, Section 5.2 demonstrates the results of every component of the algorithm. Finally, Section 5.3 analyzes the quality of the solutions regarding the operating room management.

Table 6 Comparison of the number of satisfied patients between CPLEX and our approach on benchmark B1

Id	NCP	NSP	CPLEX	Our approach	**Gap %**
1	177	354	527	527	0 %
2	204	408	532	532	0 %
3	229	458	600	616	2.67 %
4	240	479	600	616	2.67%
Average					**1.33 %**

5.1 Computational environment and benchmark instances

The technical details of the implementation of the hybrid algorithm for solving the set of developed data instances will be depicted in this subsection. In addition, we will present the set of data instances used to justify the efficiency of the suggested approach (real-life data instances). To do that, the next subsection will detail the computational environment, while the second subsection will be devoted to presenting the benchmark instances used in the experiments.

5.1.1 Computational environment

We implemented the proposed mathematical model using the IBM ILOG CPLEX Optimization Studio V12.4 [10] with default optimization parameters. CPLEX is an optimization software package that solves linearly linear programming (LP) and related problems where the objective to be optimized can be expressed as a linear function. On the other hand, we coded the three-level hybrid algorithm in the Python programming language. All tests were carried out on a Windows PC with Core i3 - 2.50 GHz CPU and 6 GB RAM.

5.1.2 Benchmark instances

To generate realistic instances which will be used to test our modeling and solving approach, we relied on real data provided by the Department of Respiratory and Chest Diseases of the King Abdulaziz Hospital, Jeddah, KSA. The provided data instances contain all the relevant information

Table 5 Comparison of solution quality between CPLEX, our approach, and the optimal solution on benchmark B1

Id	NCP	NSP	CPLEX	Our approach	Optimal solution	CPLEX gap to optimal solution	Our approach gap to optimal solution
1	177	354	1054	1054	1062	0.76 %	0.76 %
2	204	408	1064	1064	1224	15.04 %	15.04 %
3	229	458	1216	1232	1374	12.99 %	11.53 %
4	240	479	1216	1232	1438	18.26 %	16.72 %
Average						**11.76 %**	**11.01 %**

Table 7 Comparison of solution quality between CPLEX and our approach on benchmark B2

Id	NCP	NSP	CPLEX	Our approach	Gap %
5	267	533	1399	1596	14.08 %
6	340	679	1534	1868	21.77 %
7	415	829	1674	2148	28.32 %
8	479	958	1859	2459	32.28 %
Average					**24.11 %**

Table 9 Comparison of the number of satisfied patients between CPLEX and our approach on benchmark B2

Id	NCP	NSP	CPLEX	Our Approach	Gap %
5	267	533	600	798	33 %
6	340	679	600	934	55.67 %
7	415	829	600	1074	79 %
8	479	958	600	1200	100 %
Average					**66.92 %**

regarding 2500 patients, 30 nurses, a variable number of recovery rooms with respiratory machines, and a variable number of recovery rooms with recovery beds.

From this information, we generated two different benchmarks each composed of 4 instances. The first benchmark *B1* is composed of critical state patients and the second benchmark *B2* is composed of severe state patients. The set of instances have a variable number of rooms with recovery beds (RB) and a variable number of rooms with respiratory machines (RM). However, all data instances have the same number of caregivers, $N = 30$. The characteristics of the benchmark sets are summarized in the following Tables 2 and 3 in which the headings are as follows:

- B1 and B2 refer to the first and second generated benchmarks.
- ID: the number of data instances.
- NCP: number of critical state patients.
- NSP: number of severe state patients.
- nRRR: number of recovery respiratory rooms.
- nRBR: number of recovery beds rooms.
- NRM: number of respiratory machines.
- NRB: number of recovery beds.
- NRRR: nurse that serves patients of RRR.
- NRBR: nurse that serves patients of RBR.

5.2 Computational results

We develop in this subsection a set of computational experiments to show the performance of our proposed approach. Main conclusions are derived based on the computational efficiency in terms of two parameters which are:

- *The Optimality Gap:* which requires the comparison of the actual performance with the potential (CPLEX results) and the desired performance (the optima solution) [11].
- *The CPU Time:* The amount of time needed to reach the solution.

The computational results of both parameters will be reported in the following two subsections and will be illustrated by a set of graphical charts to show the difference between our proposed solution and the CPLEX solution.

5.2.1 Objective function results

We first analyze the general performance of the suggested approach methodology (the objective function to optimize). Tables 4, 5, and 6 report the results of the instances analyzed in benchmark B1. Firstly, we compare the solution quality of our proposed approach against the CPLEX solution (see Table 4). Secondly, the comparison of the solution quality between our proposed approach, the CPLEX solution, and the optimal solution (see Table 5) which can be defined as a feasible solution where the objective function reaches its maximum value. In our context, the optimal solution consists of satisfying all patients in the data instance. Finally, the comparison of the number of satisfied patients between the CPLEX solution and our approach. The same

Table 8 Comparison of solution quality between CPLEX, our approach, and the optimal solution on benchmark B2

Id	NCP	NSP	CPLEX	Our approach	Optimal solution	CPLEX gap to optimal solution	Our approach gap to optimal solution
5	267	533	1054	1399	1596	14.37 %	0.25 %
6	340	679	1064	1534	1868	32.86 %	9.10 %
7	415	829	1216	1674	2148	48.63 %	15.83 %
8	479	958	1216	1859	2459	54.60 %	16.88 %
Average						**37.61 %**	**10.51 %**

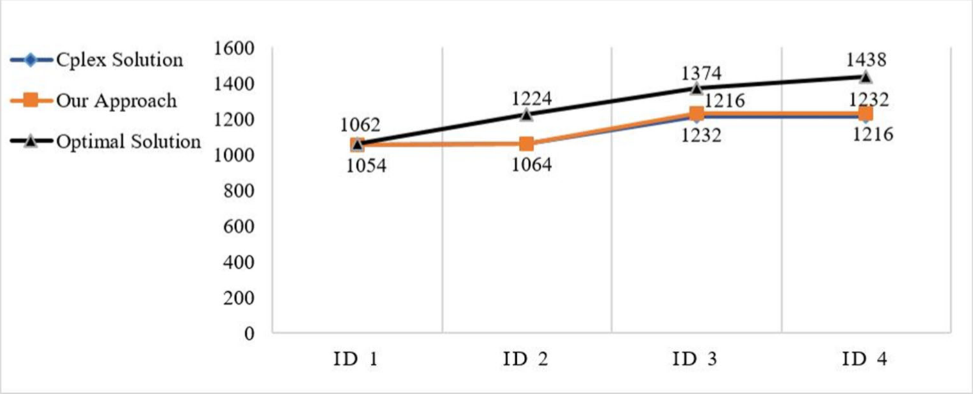

Fig. 3 Objective function comparison between CPLEX solution, our approach solution, and the optimal solution on Benchmark B1

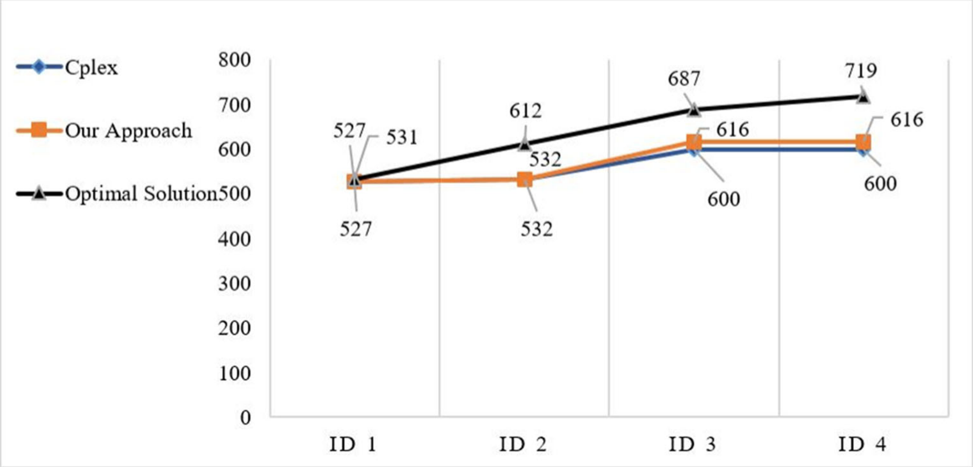

Fig. 4 Comparison of the number of satisfied patients between CPLEX solution, our approach solution, and the optimal solution on benchmark B1

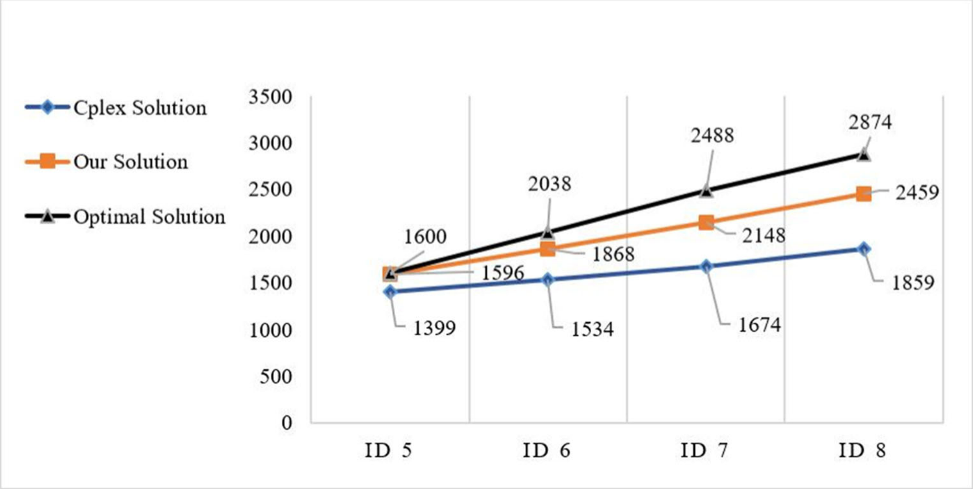

Fig. 5 Objective function comparison between CPLEX solution, our approach solution, and the optimal solution on benchmark B2

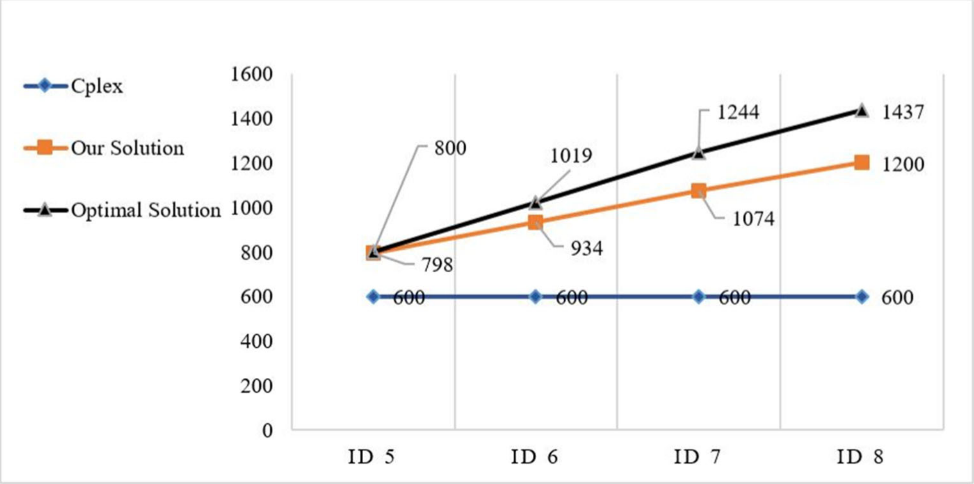

Fig. 6 Comparison of the number of satisfied patients between CPLEX solution, our approach solution, and the optimal solution on benchmark B2

sets of comparisons were provided on B2 benchmarks in respectively Tables 7, 8, and 9.

In Figs. 3 and 4, we will respectively simplify the reported results in the previous Tables 5 and 6 using two graphical charts that illustrate the comparison between (i) the solution quality (Fig. 3) and (ii) the number of satisfied patients (Fig. 4) between CPLEX solution and our approach on benchmark B1. On the other hand, the two Figs. 5 and 6 will respectively simplify the reported results in the previous Tables 8 and 9 to illustrate the same comparison between CPLEX solution and our approach on benchmark B2.

5.2.2 CPU time results

In this subsection, we present the CPU time required to reach the final solution. We have set a time limit of 1 h of CPU computational time for CPLEX to reach a final solution (i.e., 3600 s). Tables 10 and 11 report respectively the computational times in seconds for all data instances in B1 and B2 benchmarks.

In Figs. 7 and 8, we will simplify the reported results in the previous Tables 10 and 11 using two graphical charts that illustrate the computational time comparison between the CPLEX solution and our solution approach on Benchmarks B1 and B2.

5.3 Solution quality evaluation

This section will be devoted to evaluate the overall quality of our suggested approach using multiple metrics [13]. In our context, the algorithmic approach will be evaluated using essentially two measures which are:

1. *The quality of the proposed solution* (the objective function value) compared to the CPLEX solution and the optimal solution,
2. *The CPU time* required to reach the final solution.

Both previous performance criteria will be presented, illustrated, and discussed in the next two subsections by developing some statistical tests for measuring the performance of the solution. In this context, two different statistical tests will be conducted to compare our results to those obtained by the CPLEX optimizer which are (i) the sign pairwise comparison test that calculates the number of wins realized by each of the compared solutions, and (ii) Wilcoxon signed-rank test to prove that the obtained results are significant.

5.3.1 Statistical analysis of the objective function solution

It is important to compare the values of the objective function obtained by our hybrid metaheuristic approach

Table 10 Comparison of CPU computational time between CPLEX and our approach on benchmark B1

Id	NCP	NSP	CPLEX	Our approach (in seconds)
1	177	354	2327.20	33.02
2	204	408	3600	35.84
3	229	458	81.98	37.32
4	240	479	177.97	38.92
Average			**1546.79**	**36.27**

Table 11 Comparison of CPU computational time between CPLEX and our approach on benchmark B2

Id	NCP	NSP	CPLEX	Our approach (in seconds)
5	267	533	3420.11	56.86
6	340	679	3600	67.56
7	415	829	647.52	73.24
8	479	958	3600	73.20
Average			**2816.91**	**67.72**

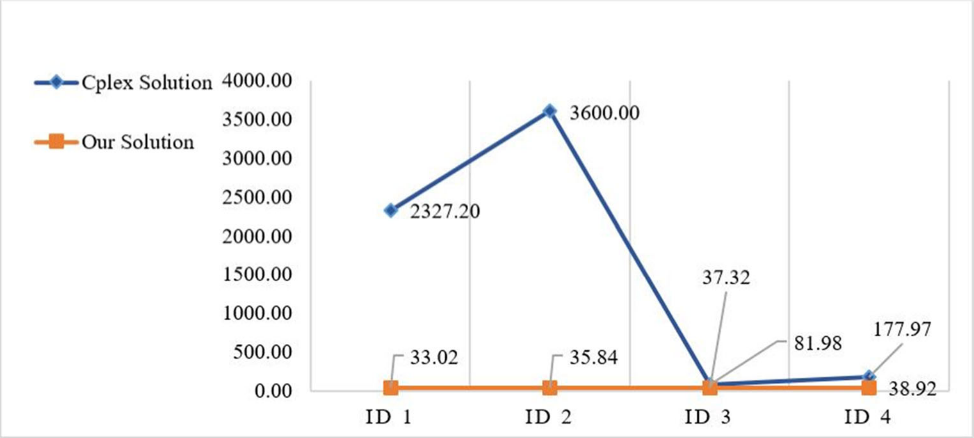

Fig. 7 Computational time comparison between the CPLEX solution and our solution approach on benchmark B1

and those obtained by the CPLEX software. To do so, we provide in the following two Tables 12 and 13 some statistical tests, to compare and analyze the performance of our proposed approach compared to the CPLEX solution on both benchmarks B1 and B2.

Then, using previous Tables 12 and 13, we conclude that there is a significant improvement of the three-level hybrid algorithm compared to the CPLEX solution. Based on the objective function feature, our solution outperforms the CPLEX solution in 2 data instances over 4 data instances of B1 benchmarks (equal results in the two other data instances). Also, our results are better than the results provided by CPLEX software in all 4 data instances of B2. This means that the efficiency of our hybrid metaheuristic increases when the size of the problem grows (the number of patients in B2 is larger than the number of patients in B2), thus indicating that effective solutions can be found even for large instances.

In addition, based on the detailed results in Table 4, our approach results are better than CPLEX results by 0.66% on average for the four first data instances. This improvement can be considered low due to the size of the used data instance. On the other hand, the gap increases when the size of the data instances increases, i.e., the results provided in Table 7 indicate that our solution outperforms the CPLEX results by 24.11% on average. The same interpretations can be extracted when we conduct our results analysis on comparing the number of satisfied patients between CPLEX and our approach on both benchmarks B1 and B2 (see respectively Tables 6 and 9).

Thus, we can interpret that our obtained results outperform those obtained by the CPLEX results in 6 over 8 data instances.

5.3.2 Statistical analysis of the CPU computational time

Generally, when solving a combinatorial optimization problem, we look to find the best trade-off between the solution quality and the running time expended to having it. So, decreasing the computational time is an important evaluation measure to prove the high quality of the proposed solution. In this subsection, we will present, evaluate, and compare the computational time needed by different approaches to reach a near-optimal solution for all

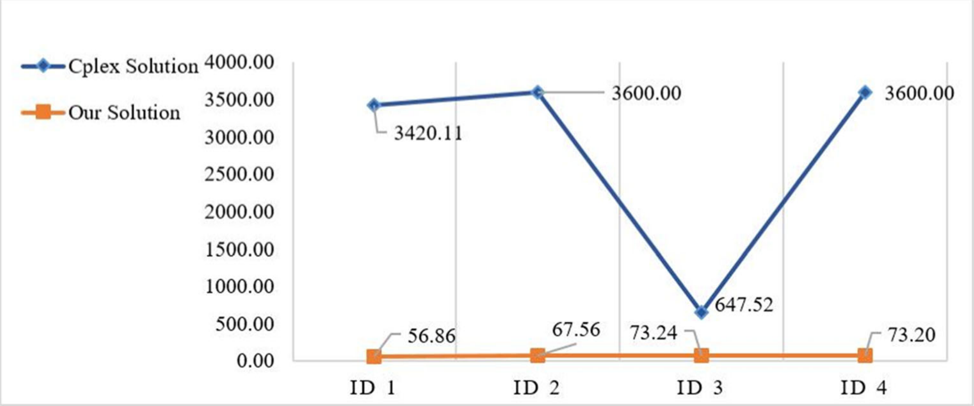

Fig. 8 Computational time comparison between the CPLEX solution and our solution approach on benchmark B2

Table 12 Sign test for pairwise comparisons of the objective function on B1 benchmark

	CPLEX	Our approach
Wins	0	2
Losses	2	0
Equal	2	2

Table 14 Sign test for pairwise comparisons of the computational running time on the benchmark B1

	CPLEX	Our approach
Wins	0	4
Losses	4	0
Equal	0	0

data instances. In this context, [5] defines a near-optimal solution as "a feasible solution with an objective function value within a specified range from the (usually unknown) optimal objective function value."

To do so, we provide in the following two Tables 14 and 15, two statistical tests to analyze and compare the CPU time used by our proposed approach compared to the CPLEX solution on both benchmarks B1 and B2.

According to both Tables 14 and 15, we can remark a significant improvement of our proposed solution over the CPLEX solution based on the required CPU time to reach a final solution since our approach outperforms the results obtained using CPLEX software in all 8 data instances B1 and B2 benchmarks.

In addition, the detailed results given in Tables 10 and 11 show a huge improvement in the time needed by the CPU to reach a near-optimal final solution. Our approach only needed 36.27 s on average for the four first data instances against 1546.79 s on average for the CPLEX optimizer. Also, for the second benchmark, our approach needs only 67.72 s to reach a near-optimal solution against 2816.91 s for the CPLEX approach.

Moreover, both Figs. 7 and 8 show that the CPU time required by our algorithm to find a near-optimal solution is optimized compared to the CPLEX approach.

5.3.3 Wilcoxon signed-rank test

In order to judge that our obtained results are significant and not happened by chance, a nonparametric statistical test is required to be accomplished. In this context, we choose to develop Wilcoxon's rank-sum test which is a nonparametric statistical to justify that the significant performance is not happened randomly based on the p-values [3]. In our study, the test hypothesis are:

Null Hypothesis H0: The number of satisfied patients is equal to the number of satisfied patients of CPLEX solution.

Alternative Hypothesis H1: The number of satisfied patients is equal to the number of satisfied patients of CPLEX solution.

The p-values reached when conducting the Wilcoxon signed-rank test on the four data instances of B1 and the four data instances of B2 in addition to the standard deviation values are illustrated respectively in Tables 16 and 17. Each table presents the related statistical results obtained by our approach compared to CPLEX results for each benchmark independently.

According to results in both previous tables, we have to reject the null hypothesis (no significant difference between our results and CPLEX result) and accept the alternative hypothesis which ensures that our proposed approach is significantly better than the CPLEX results. Thus, the Wilcoxon signed-rank test proves that our solution definitely shows a general improvement over the CPLEX solution based on the main comparative features, which are the number of satisfied patients and the needed CPU to reach a final solution.

5.3.4 Study limitations

Although results show that our proposed modeling and solving approach significantly outperforms the results obtained by CPLEX software, it is mandatory to mention that there are some limitations of our study that must

Table 13 Sign test for pairwise comparisons of the objective function on B2 benchmark

	CPLEX	Our approach
Wins	0	4
Losses	4	0
Equal	0	0

Table 15 Sign test for pairwise comparisons of computational running time on the benchmark B2

	CPLEX	Our approach
Wins	0	4
Losses	4	0
Equal	0	0

Table 16 The Wilcoxon signed-rank test of the number of satisfied patients

Benchmark	Number of instances	+ Ranks	- Ranks	Standard	*p*-value	Decision
B1	4	2	0	1.061	**0.0157**	Reject the null hypothesis
B2	4	4	0	2.739	**0.0168**	Reject the null hypothesis

be mentioned here, from which we cite some in the following:

– Our approach does not consider the time taken for the administrative procedure when receiving a new patient. The reception procedure asks for numerous information about the patients especially in the case of COVID-19 such as the age, the body mass index (BMI), the used medications (if exist), and any historical diseases. This waste of time may result in a delay that could affect the scheduling process.
– The actual approach considers only the deterministic variant of the problem. However, our approach cannot handle the stochastic variant of the problem which consists of non-deterministic (real-time) data instances.

6 Conclusion

In this paper, we have addressed the hard problem of planning and scheduling integrated recovery rooms during the COVID-19 pandemic. To answer this urgent concern of healthcare providers, the IRRPSP is defined and then solved by decomposing it into two sub-problems which are solved in three sequential phases. In the first step, the method search to assign patients to the recovery rooms while respecting both types of patients and recovery rooms. The method, therefore, seeks to assign nurses to patients. In this second step, we used the overtime resources in order to maximize nurses' utilization while minimizing the number of unsatisfied patients. The hybrid algorithm developed combines neighborhood search techniques with the Tabu Search metaheuristic, which is used to optimize the number of satisfied patients. The proposed approaches have been explained, justified, and illustrated with synthetic and pedagogical figures and diagrams. Then, they have been programmed and tested on a set of real data that have been constructed. The reported results mainly achieved objective function levels and the needed computational time shows the efficiency and the robustness of the solutions produced.

Two main and significant outcomes of this work are to point out, from our standpoint. The first is the tackling of such interesting subjects especially there are no sufficient works that try to develop solutions to COVID-19 pandemic problems. Moreover, with the increasing challenges faced by the health sector especially during the COVID-19 virus, this topic gets more and more important and needs to be addressed and solved.

The second interesting point revealed by this work is the result performance against the CPLEX results. The computational analysis demonstrated the capability of the suggested approach to deal efficiently with the trade-off between the available nurses and patient perspectives in reasonable computational times.

The perspectives we intend to give to this research work are threefold. In the first direction, we are particularly interested in computing the strategies developed in this paper with bigger data instances. The second direction in which we are interested is to continue our research work based on these promising results is to implement more sophisticated heuristical algorithms, relying on the proposed pattern decomposition. The third perspective consists of transforming the proposed three-level hybrid algorithm into a real-life scenario by applying it to a practical case for one of Jeddah hospitals. The main steps of this scenario can be captured in Fig. 2, in which the first task consists of assigning the patients to the recovery rooms then the second task takes the output of the previous task as an input to define the nurses' assignments to patients. A third step of the scenario consists of improving the obtained solution of the second step by minimizing the set of unsatisfied tasks. This can be done by (i) profiting from the available overtime of the nurses and (ii) by maximizing the recovery room utilization.

Table 17 The Wilcoxon signed-rank test of the CPU time

Benchmark	Number of instances	+ Ranks	- Ranks	Standard error	*p*-value	Decision
B1	4	4	0	3.921	**0.026**	Reject the null hypothesis
B2	4	4	0	1.326	**0.052**	Reject the null hypothesis

Funding Dr. Marouene CHAIEB is a recipient of a grant from the Scientific Research Deanship at Sattam bin Abdulaziz University under the project number 2021/01/18367.

Declarations

Ethics approval The study conformed to the institute's ethical standards.

Competing interests The authors declare no competing interests.

References

1. 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 p 1–24
2. Bouguerra A, Sauvey C, Sauer N (2020) A decision support tool for the urgent surgeries assignment problem. Journal of Engineering Research 8(2)
3. Eftimov T, Korošec P (2019) A novel statistical approach for comparing meta-heuristic stochastic optimization algorithms according to the distribution of solutions in the search space. Inf Sci 489:255–273
4. Eun J, Kim SP, Yih Y, Tiwari V (2019) Scheduling elective surgery patients considering time-dependent health urgency: Modeling and solution approaches. Omega 86:137–153
5. Gass SI, Harris CM (1997) Encyclopedia of operations research and management science. J Oper Res Soc 48(7):759–760
6. Glover F (1989) Tabu search - part i. ORSA J Comput 2(1):190–206
7. Güler MG, Geçici E (2020) A decision support system for scheduling the shifts of physicians during covid-19 pandemic. Comput Industrial Eng 150:106874
8. Hamid M, Nasiri MM, Werner F, Sheikhahmadi F, Zhalechian M (2019) Operating room scheduling by considering the decision-making styles of surgical team members: a comprehensive approach. Comput Oper Res 108:166–181
9. Hashemi Doulabi SH, Rousseau LM, Pesant G (2016) A constraint-programming-based branch-and-price-and-cut approach for operating room planning and scheduling. INFORMS J Comput 28(3):432–448
10. ILOG I (2012) Cplex optimizer. En ligne] Available: https://www.ibm.com/analytics/cplex-optimizer
11. Kim S, Ji Y (2018) Gap analysis. The International Encyclopedia of Strategic Communication p 1–6
12. Ktari R, Chabchoub H (2013) Essential particle swarm optimization queen with tabu search for mkp resolution. Computing 95(9):897–921
13. Li M, Yao X (2019) Quality evaluation of solution sets in multiobjective optimisation: a survey. ACM Computing Surveys (CSUR) 52(2):1–38
14. Luo YY, Wang B (2019) A new method of block allocation used in two-stage operating rooms scheduling. IEEE Access 7:102820–102831
15. M'Hallah R, Visintin F (2019) A stochastic model for scheduling elective surgeries in a cyclic master surgical schedule. Comput Ind Eng 129:156–168
16. 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. Comput Ind Eng 80:72–79
17. Tomasic I, Tomasic N, Trobec R, Krpan M, Kelava T (2018) Continuous remote monitoring of copd patients justification and explanation of the requirements and a survey of the available technologies. Med Biol Eng Comput 56(4):547–569
18. University JH (2021) Covid-19 dashboard by the center for systems science and engineering (csse). Computers & Operations Research, https://coronavirusjhuedu/maphtml
19. Worldometers (2021) Covid-19 coronavirus pandemic. https://www.worldometersinfo/coronavirus/

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Marouene Chaieb is an assistant professor of the Information Systems Department at Prince Sattam bin Abdulaziz University, KSA. He holds a bachelor in computer science, a masters degree in information systems, and a Ph.D. in information systems from the Higher Institute of Management of Tunis, Tunisia. He is the head of Statistical and Analysis Department in Prince Sattam bin Abdualziz University. His areas of interest and research are in the healthcare service delivery, systems modeling and optimization, logistics and supply chain management, medical computing, and decision making. He is a member of the LARODEC laboratory from 2006.

Dhekra Ben Sassi is an assistant professor of the Information Systems Department at Prince Sattam bin Abdulaziz University. She holds a bachelor's in computer science, a master's degree in information systems, and a Ph.D. in information systems from the Higher Institute of Management of Tunis, Tunisia. Her areas of interest and research are in business intelligence, optimization, and decision support systems. She is a member of the RIADI laboratory since 2007.

Jaber Jemai holds a Bachelor, Master, and PhD degrees in Computing Information Systems from the University of Tunis and the postgraduate diploma in Data Science and Business Analytics from Mc Combs School of business, University of Texas at Austin. He justifies more than 19 years of experience in academia. He is currently associate professor in Computer and Information Systems division of the Higher Colleges of Technology in the UAE. The research interests of Dr. Jaber include combinatorial optimization and machine learning techniques and their applications to solve logistics, transportation, healthcare, financial problems among others. He authored and co-authored more than 20 papers published in international refereed journals and conferences.

Khaled Mellouli was born in Tunisia. He holds a BSc in computer science, a MSc in computer science, and a Ph.D. from the USA. His main research interest is in the area of artificial intelligence (AI) and optimization problems (OP). Actually, a Full Professor at Institut Haute des etudes commerciale (IHEC Tunis) and the director of LARODEC Laboratory, Institut Superieur de Gestion de Tunis.

Affiliations

Marouene Chaieb[1,2] · **Dhekra Ben Sassi**[1,3] · **Jaber Jemai**[4] · **Khaled Mellouli**[2]

Dhekra Ben Sassi
d.bensassi@psau.edu.sa; bensassi.dhekra@yahoo.fr

Jaber Jemai
jaber.jemai@gmail.com

Khaled Mellouli
mellouli.khaled@gmail.com

[1] College of Computer Engineering and Sciences,
Prince Sattam bin Abdulaziz University,
Al-Kharj, Building N3963, 16273, Saudi Arabia

[2] LARODEC, Institut Supérieur de Gestion de Tunis,
Université de Tunis, 41 Ave de la Liberte, Tunis 2000, Tunisia

[3] RIADI, École Nationale des Sciences de l'Informatique
(ENSI), Campus Universitaire de la Manouba,
2010 Manouba, Tunisia

[4] Computer and Information Systems Division, Higher Colleges of
Technology, Abu Dhabi, United Arab Emirates