Omega 123 (2024) 102994

Contents lists available at ScienceDirect

Omega

journal homepage: www.elsevier.com/locate/omega

Appointment scheduling in surgery pre-admission testing clinics

Saligrama Agnihothri [a], Paola Cappanera [b,*], Maddalena Nonato [c], Filippo Visintin [d]

[a] School of Management, Binghamton University, Binghamton, NY, United States
[b] Dipartimento di Ingegneria dell'Informazione, Università degli Studi di Firenze, Firenze, Italy
[c] Dipartimento di Ingegneria, Università degli Studi di Ferrara, Ferrara, Italy
[d] Dipartimento di Ingegneria Industriale, Università degli Studi di Firenze, Firenze, Italy

ARTICLE INFO

Keywords:
Appointment scheduling
Pre-admission testing
Machine scheduling
Matheuristics

ABSTRACT

Pre-admission testing clinics are care units serving outpatients prior to surgical operation and performing procedure-specific tests to prepare them. Patients may need multiple tests, each performed by a specialized operator and delivered in any order. Exam rooms act as renewable resources: rooms are limited, tests are administered to patients inside the rooms, individually, and patients occupy the room until all the required tests are completed.

Careful scheduling of patient appointments is essential in clinic management for both the patient and the provider: on the one hand, minimizing patient waiting time improves service quality, on the other hand, minimizing completion time (makespan) improves system efficiency.

In this paper, we propose offline policies for the daily scheduling of pre-admission test appointments. As a benchmark, we consider two online scheduling policies widely used in common practice. Each of these offers a different compromise between complexity and resource exploitation.

The proposed optimization-based offline booking policy is identified as a new problem in the machine scheduling literature, for which we propose a network-flow model representation. A family of matheuristics based on different variable fixing criteria is provided to circumvent the high computational effort required to solve the mathematical model to optimality on real-size instances. The performance, advantages and disadvantages of each of the online and offline policies are compared in a variety of scenarios based on realistic data.

Through this work, decision-makers have a new set of tools they can choose from according to their priorities.

1. Introduction

Surgical outpatients undergo standardized medical tests shortly before their procedure to assess eligibility. Pre-Admission Testing (PAT) clinics devoted to this purpose have been in practice for the last 20 years and proved to be effective in minimizing surgery date cancellations [1,2], reducing post-operative length of hospital stay, or even avoiding it [3].

In practice, PAT can be implemented in two different manners, either distributed or centralized. In the former, service delivery is fragmented: patients move individually across different hospital departments, each providing a different kind of test, and patients queue for service at each facility [4]. In the latter, the service is centralized at a single facility, typically an outpatient clinic, where different operators gather to provide the needed services in a seamless manner [5]. The advantages of this approach are many: patients waste no time in moving around different locations; patient data are centrally managed and duplicate queries are avoided; different providers can interact and have a holistic view of patients. In this paper, we address a patient-oriented variant of the second option.

This study was inspired by a real PAT clinic in a hospital, the details of which are given in Appendix A. We summarize important features of the problem hereafter. Patients may take multiple tests, each performed by an operator with a specialized skill. Tests can be administered in any order. At the clinic, there are a limited number of exam rooms. Once patients are taken to a room, one at a time, they remain inside it until all the required tests are completed. Color-coded flags above the door of each busy exam room show in real time which tests are still to be performed on the patient currently inside, enabling a free operator

* Corresponding author.
E-mail addresses: agni@binghamton.edu (S. Agnihothri), paola.cappanera@unifi.it (P. Cappanera), maddalena.nonato@unife.it (M. Nonato), filippo.visintin@unifi.it (F. Visintin).

https://doi.org/10.1016/j.omega.2023.102994
Received 23 March 2022; Accepted 28 October 2023
Available online 7 November 2023

to know in which rooms their service is needed. Thus, a patient may have to wait in the exam room for an operator to perform a specific test, or an operator may have to wait if the patient is undergoing a test performed by a different operator. Test duration may vary but depends on the test rather than on the patient. Some operators can be multiskilled (cross-trained).

In this paper, we address a simplified though realistic variant of the above, which we call PAT Appointment Scheduling (PAT-AS) problem. Specifically, PAT-AS consists of scheduling a working session at a PAT clinic with single-skilled operators, where (i) requested service is known at booking time; (ii) test variety is limited and much lower than the number of patients attended during one working session; (iii) tests are standardized and test duration is deterministic; (iv) a patient undergoes multiple tests (a test package) in a single session, in any order; (v) each patient is served by a skilled operator at a time and vice versa; (vi) each operator can perform only one kind of test; (vii) each working session can be scheduled independently since patients are not expected to ask for a second appointment within the same health pathway; (viii) once a room has been assigned to a patient, it will be released only after the patient has undergone the whole test package (*the room constraint*). For a given resource configuration, namely, a fixed number of rooms and operators, PAT-AS consists of (a) assigning patients to rooms, (b) scheduling patients assigned to the same room, (c) sequencing each patient's tests, and (d) planning the sequence of tests performed by each operator, accordingly. Solution performance indicators considered in this study are the makespan, defined as the total time required by the system to process all requests, and the total time patients spend inside a room waiting for an operator. Despite the simplifications, in particular with regard to operator skills, we claim that understanding PAT-AS is an unavoidable step towards the solution of more general versions and argue that PAT-AS is a problem of interest on its own.

The main contribution of this paper is to assess the pros and cons of introducing the room constraint into a scheduling problem with three different booking policies, namely, walk-in, slot-based, and offline optimization-based (OPT) ones. The OPT policy identifies a new problem in the machine scheduling literature. We highlight the impact of rooms as shared resources and show why greedy algorithms from the literature may not perform as expected when solving PAT-AS. A network-flow-based mathematical model is presented to support the OPT policy, where the bottleneck operator scheduling is recognized as the core decision. Based on that, three effective and efficient variants of a matheuristic algorithm are proposed. Extensive computational results on generalized real-size instances prove that one of them achieves near-optimal solutions in limited computational times, even when relaxing some features of the original problem.

The paper is organized as follows: Section 2 provides a formal problem description. Related papers are reviewed in Section 3 to support the claim that we are dealing with a new problem. Section 4 presents online and offline policies, the mathematical model supporting the OPT policy, and the matheuristics. These are tested against different scenarios in Section 5, where results are reported and discussed. The managerial insights obtained from this study are presented in Section 6, followed by final conclusions reported in Section 7.

2. Problem features

We first formalize the primary features of PAT-AS. Table 1 provides the main symbols and parameters of the mathematical notation.

We assume the following: (1) Both human and material resources are fixed (no resource sizing or sharing) and decisions only concern the schedule. (2) All patients in the patient's set P have equal priority. (3) P is partitioned into classes where class $P_c \subseteq P$ corresponds to test package c; patients in the same class are identical but patients in different classes ask for different packages. (4) Tests in set T can be taken in any order. (5) The duration d_t of each test $t \in T$ is deterministic and does not depend on the patient. (6) Planning horizon spans one working shift of duration H minutes. Service policy is to serve everybody within H in the offline policy, anyone arrived either within H or within $H - \Delta$ in the two online policies, where Δ is the duration of the longest test package a patient in P may take. (7) There is one qualified operator o^t for each test t and each operator is qualified for a single test (no cross-training, one-to-one correspondence between tests and operators). (8) A fixed number n_R of identical exam rooms are available and this is equal to the number of different tests ($n_T = n_R$). Rooms are available during the whole planning horizon. (9) Scheduled patients always turn up and arrive on time to scheduled appointments. The assumptions listed so far result in the following set of constraints: (a) One patient per room at a time. (b) No room pre-emption (according to the room constraint, patients release the room only after all their tests have been completed). (c) Each operator must serve one patient at a time and each patient cannot be visited by more than one operator at a time; tests must be completed without interruption. (d) According to assumption (4), the order of a patient's tests is immaterial.

Table 1
Main symbols and parameters. For any patient $p \in P_c$, δ^p is equal to σ_c: the most suitable notation will be used, depending on the context.

H	Planning horizon length
P	Set of n_P patients
T	Set of n_T tests
d_t	Duration of test $t \in T$
$T^p \subseteq T$	Patient p's package
$\delta^p = \sum_{t \in T^p} d_t$	Service time for patient p
$P(t) = \{p \in P \mid t \in T^p\}$	Set of patients with test t in their package
$\Delta = \max_{p \in P}\{\delta^p\}$	Duration of the longest test package taken by the patients in a given set P
$C = \{c \subseteq T \mid \exists\, p : c = T^p\}$	Set of all packages
$P_c = \{p \in P \mid T^p = c\}$	Patients with the same package c
$\sigma_c = \sum_{t \in c} d_t$	Service time of test package c
o^t	The one operator qualified to deliver test t
$O = \{o^t \mid t \in T\}$	Set of $n_O = n_T$ operators
R	Set of n_R rooms

The peculiar features of this problem are (a) and (b), given (4) and (5). Constraint (a) imposes a maximum parallelism, i.e., at any time there are at most as many patients being served (active operators) as the number of rooms. Due to constraint (b), when a patient is idle inside a room, waiting to be served by a busy operator, patients waiting outside the room to be served cannot be served by idle operators due to a lack of free rooms. This phenomenon is called *non-work conserving service discipline* in queuing literature. This inefficiency, that we call a *temporary deadlock*, would be avoided if rooms were released after each test (room preemption). In such a case, rooms would simply impose a maximum parallelism on the number of operators active at the same time, which is not binding in the $n_R = n_O$ case. On the other hand, the room constraint is stronger than a bare maximum parallelism among operators, and it is tight even for $n_R = n_O$. For a given schedule, let us introduce $psp(p)$ as the *patient service period* of patient p, i.e., the smallest time interval in which p receives the required test package. Now, the room constraint sets n_R as an upper bound on the number of *psps* with mutual intersection, as the next example shows.

Consider the case of (i) $n_R = n_O = n_T = 2$, (ii) 3 patients p_1, p_2, p_3 requiring both tests each, (iii) service times d_1 and d_2 such that $d_1 > d_2 > d_1/2$. Clearly, $3d_1$ is a lower bound for the makespan. First, suppose the room constraint is enforced. Then, the optimal makespan is $2(d_1 + d_2) > 3d_1$ and patients can be served with no waiting time as follows; first, o^1 serves p_1 in room one while o^2 serves p_3 in the other room, starting at time $d_1 - d_2$. At time d_1 the operators switch room: o^1 serves p_3 in room two while o^2 serves p_1 and then p_2 in room one. Finally, at time $d_1 + 2d_2$, o^1 is back to the first room to serve p_2 and the process ends at time $2(d_1 + d_2)$. At most two *psps* have mutual intersection. This solution is depicted in Fig. 1, on top, where t_1 is the blue test and t_2 the yellow one, with $d_1 = 5$ and $d_2 = 3$. Now,

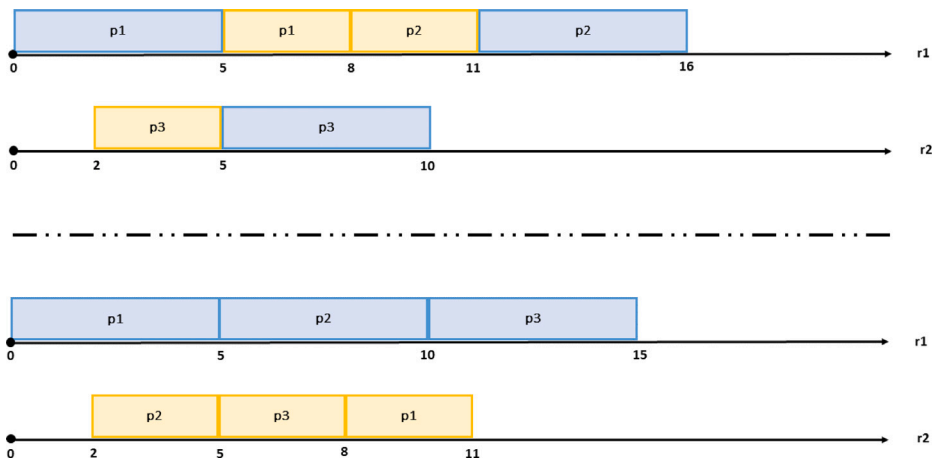

Fig. 1. A solution complying (top) and not complying (bottom) with the room constraint. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

let us allow room preemption, i.e., the room is released after each operation and a patient may take different tests in different rooms. Indeed, o^1 could serve p_1, p_2, p_3 in the first room in this order, while o^2 could serve p_2, p_3 and p_1 in this other order in the other room. The resulting makespan is $3d_1$. However, now the *psp*s of the three patients are mutually intersecting, such as at time $t = d_1 + d_2$, even though at most two patients are being served at a time. This solution is depicted at the bottom of Fig. 1.

The room constraint may further affect a schedule's performance. In particular, optimal schedules may not be *dense* (in a dense schedule, an operator is idle if and only if there is no idle patient still in need of that service). In fact, for $n_T = n_R$, whenever a patient is idle inside a room, one operator is idle at the same time; likewise, any other patient not yet admitted to a room and in demand for that same operator is stuck even though that operator is available. This temporary deadlock would not take place without the room constraint. All these considerations bring to the forefront the centrality of scheduling policies in the PAT-AS framework.

3. Literature

Literature on Outpatient Appointment Scheduling (OAS) is briefly revised to point out the peculiar features of PAT-AS (Section 3.1). Then, the main results in the machine scheduling literature are presented and PAT-AS is restated in that framework, to support the claim of its novelty (Section 3.2).

3.1. Outpatient appointment scheduling

PAT appointment scheduling is part of the larger family of Outpatient Appointment Scheduling (OAS) but differs from most of them because of its own specificities.

Papers in this field can be classified according to:

1. Source of variability in the input parameters. The most common one concerns demand. Variability may concern service time, patient punctuality, no-shows, and walk-ins, the first one being the most addressed one; similar sources of uncertainty may arise on the resource side, regarding service providers' punctuality.
2. Single-stage or multi-stage service. Demand consists either of a single service request (single-stage) or of a bundle of requests (multi-stage), each one delivered by one (or more) specific service provider(s). In the latter case, the sequence in which stages are delivered (patient flow) may be fixed or not. As far as the time horizon is concerned, service may have to be delivered during a single visit, i.e., on the same day, or during recurring visits (with potential time constraints between appointment dates). Many of the studies address single-server, single-stage processes.
3. Solution quality criteria (what the authors wish to pursue), potentially conflicting. One measure of service quality is patient waiting time. This may refer to (i) *indirect waiting time*, the time between the day the request is made and the appointment date, (ii) *direct waiting time*, the waiting time in the clinic before the patient is served, or (iii) the time spent waiting between successive services, as we see in this study. A second measure concerns system efficiency in terms of resource utilization, typically minimizing operators' idle time and overtime, or makespan (alternatively maximizing patient throughput). Indeed, an increasingly patient-centered perspective means that the focus shifts to quality of care rather than cost of the service. However, inefficiencies such as operators' idle time can lead to operational losses. Therefore, a balance between service quality and system efficiency is usually sought.
4. The kind of schedule to be produced as well as the process that leads to that schedule. When the schedule is determined by sequencing and appointment rules in an online oriented perspective, the performance is typically assessed by simulation. Conversely, the schedule may consist of a timetable where each patient has a different appointment time that is computed offline, by solving an optimization problem.

The study in [6] considered two main classes within the online oriented policies. The first class of policies focuses on sequencing, where the aim is to give individual patients a specific appointment time by exploiting all kinds of information available to characterize the single patient. In the second class, efforts are devoted towards the definition of the appointment rule: the working period is partitioned in slots, whose duration and number of patients per slot (block size) depend on the appointment rule. Indeed, [6] compared the performance (in terms of idle time, waiting time, and overtime) of 314 appointment rules in a variety of settings by way of Data Envelopment Analysis, in the case of single server and i.i.d. service time. The study confirms the validity of the classic Bailey rule in case of limited service time variability and when there is little emphasis on waiting time. Other individual rules (1 patient per slot) perform just as well, while the block rules (multiple patients per slot) perform poorly. The dome-shaped form for slot duration was able to edge against dynamic environments.

The slot idea goes back to the pioneering works in [7] (from which the *Bailey's rule*), followed by [8,9], that first proposed to dimension the slot duration as the expected service time. Regarding sequencing rules, the basic one books the earliest vacancy in a FIFO order, while other rules assign early or late slots based on patients classification. In particular, [10] adjusted the patient sequencing rule to take into account the features of different patient classes (such as the new versus

returning dichotomy) and test it in different environments regarding service time variability, patients unpunctuality, no-show, and walk-in rates, while [11] exploited patient classification for shaping both the sequencing rule as well as the appointment rule, adjusting appointment intervals to match the consultation time characteristics of different patient classes. Walk-in patients and appointment patients are considered jointly also in [12] which addressed the case of a public hospital in Shanghai.

Other studies devise the schedule by way of optimization algorithms. One such approach is proposed in [13], which considered stochastic service time and unpunctuality. The schedule is obtained by a local search which exploits the capability of computing the solution quality of a certain schedule by assuming the knowledge of patient classification which, in turn, is related to service time. Similarly, [14] adopted a local search to optimize the weighted average of expected waiting times of patients, idle time of the doctor, and overtime, assuming that service times are independent and exponentially distributed, and patients arrive on time, but no-shows may arise. The method starts from a feasible schedule and exploits the multimodularity of the objective function to assess optimality. It builds on the findings in [15] that derived upper and lower bounds for the optimal schedule based on submodularity. More recently, [16] encompassed many sources of variability and determined the block size that optimizes a weighted sum of waiting time and idle time for a variable-sized multiblock appointment system with random service duration and time-varying no-shows, by heuristically solving a stochastic integer programming. Other optimization-based approaches, just to mention a few, are [17] which solved a two-stage stochastic linear programming problem, [18] that also considered a stochastic linear program and developed a fast heuristic for finding dome-shaped inter-arrival times, [19], whose approach is based on dynamic programming, and [20] which proposed a branch-and-bound method to find the optimal schedule although it does not scale well on large instances. More recently, [4] heuristically solved the problem of redesigning the centralized appointment system of a University Hospital. The target is to optimize patients' check-up pathway to reduce the percentage of hospitalized patients on behalf of outpatients. To this aim, the tasks of each patient are scheduled as close in time as possible, considering that (i) examinations of different categories may be delivered at distant departments, (ii) precedence constraints between tests may arise, and (iii) test duration is patient dependent.

Optimization and simulation were jointly used in [21] to solve a static, single server problem, where appointments are made in advance. Solution quality is evaluated by simulation, while a heuristic search (based on scatter search and tabú search) determines new input values for the simulation to run. Simulation optimization-based approaches are not affected by the limitations that some optimization-based approaches may suffer regarding distribution laws that model stochastic parameters or weak scaling capabilities regarding number of patients or number of stages. A simulation–optimization approach was used also in [22], where a single stage service is split into a multi-stage one by adding a mid-level service provider who attends the patient before the clinician, yielding a fixed patient flow. This introduces patient waiting time between the two stages which adds to the time spent waiting for the first stage. The paper discusses how to adjust single-stage scheduling policies to a multi-stage environment. Simulation was also used to compare fixed-length block scheduling policies commonly used in outpatient clinics, with neural network-based allocation methods, in which service time is predicted when patients call for an appointment and slot length is determined accordingly [23].

When patients need complex services, such as a mix of diagnostic tests and consultations delivered by different providers during independent sessions, or such as brachytherapy, which requires considering the decay of the radioactive source between consecutive treatments [24], the issue of indirect waiting time arises [25,26], beside possibly concentrating different appointments on the same day to reduce hospital access [27]. Actually, for specific classes of patients following a given care pathway spanning a medium-term horizon, the main issue is how to schedule multi appointments so as to respect the required time gaps between successive appointments, when resources are limited [28,29].

A broader perspective on OAS can be found in extensive literature reviews such as [30,31], covering OAS for primary and specialty care, and addressing multi-appointment services [32].

The quality of service from the patient's point of view has been addressed in several studies. For example, walk-in clinics operating on a fee-for-service basis and providing one-stage, one-server visits are dealt with in [33], which discusses whether revenue maximization pursued through increasing the number of patients seen can lead to a reduction in the duration of visits and whether minimum standards need to be enforced. The issue is controversial since longer visits tend to increase waiting times in the waiting room. The authors compare different service models and develop threshold values under which regulation is advocated. Ferreira et al. [34] measured patients' satisfaction according to Multicriteria Satisfaction Analysis (MUSA) in secondary health care-based medical appointment services provided by the NHS in Portugal and discuss applicable strategies to improve the performance of each criterion.

OAS problems that involve resource seizing can be found in outpatient chemotherapy departments [35], where an oncologist, a nurse, and an infusion chair are needed at the same time to treat the patient, in addition to pre- and post-procedure rooms. However, this problem differs from ours in that the sequence of the operations is fixed and the duration of treatment, which is the bottleneck operation, highly varies with the patient.

A similar but more articulated service configuration was described in [36] regarding an Integrated Practice Unit (IPU), where a multidisciplinary team (made of 5 different provider types) delivers treatment for joint pain during a single patient visit. IPUs share with our problem the room constraint feature. The patient flow is almost fixed, while patient dependent stage duration is a source of uncertainty. A MILP model for the deterministic case is proposed, which differs from the one we introduce in Section 4.2 but for the room constraint representation. As the model can be solved just for a few patients, the authors tackle a relaxation obtained by removing the room related constraints as well as the first step of the pathway (the nurse practitioner). The resulting problem is a hybrid flow shop, with 4 machines and one machine per operation. The simplified deterministic model is solved for a set of scenarios using expected service times to set the block size of a slot-based policy for a real case with 7 rooms.

With regard to existing PAT clinics, we mention three papers that put forward the potential as well as the organizational challenges posed by the process: [5] considers patient workflow at the Anesthesia Preoperative Clinic of the University Hospital of San Antonio, Texas. Improvements in patient care, cost savings due to better operating room utilization, and decreased unnecessary patient testing are achievable when shifting from a traditional system to an integrated PAT clinic. Results have been assessed by extensive simulation. In addition, [37,38] address resource management in an established PAT center processing up to 17 000 patients a year. The studies aim at simulating the system to identify criticalities. The findings identified the exam rooms as the most critical resource.

The study in [39] addresses the effect of coupling appointment rules with capacity allocation, intended as the number of exam rooms at an orthopedic outpatient clinic where two possible patient flows are considered, according to the outcome of the first consultation. The two flows differ in the presence of the X-ray exam that precedes entry into the exam room where the patient is seen first by the nurse and then by the physician. Simulation shows that physician idle time decreases at a decreasing rate as more exam rooms are available, from one to three rooms where the value gets almost stable. Additional exam rooms over three, reduce patient waiting time outside the room but increase waiting time in the exam room.

Appropriate patient-physician matching is becoming increasingly popular in specialty care as a means of improving care effectiveness. Recently, matching and appointment scheduling problems have been addressed simultaneously in a stochastic environment in [40].

The many sources of variability examined so far are not present in the problem addressed in this study. We assume that service time does not depend on the patient and no show and unpunctuality are almost negligible. In fact, a PAT session is very close in time to the surgery date and this time gap is correlated with non-compliant behaviors [41]. Nevertheless, despite a deterministic environment, the problem does not necessarily boil down to an easy task. We will show that in the presence of multi-stage service and tight constraints on human and material resources, the resulting scheduling problem may be challenging. Under very strong assumptions the problem can be easy to solve. In particular, in [42] polynomial time algorithms are given for two special cases, provided that, for each test, the number of identically skilled operators is such that the ratio of operation service time over the number of operators is the same for all operations. The result is weakened by the very strong hypothesis that all patients take the full set of tests, assumed also in [43]. On the contrary, we deal with several classes of patients, each characterized by a subset of tests and a specific service time. Therefore, despite many common points, none of the works that we are aware of in OAS addresses the same problem we are tackling. Even though multistage problems have been addressed, they usually involve fixed patient flow and/or the room constraint is not addressed. In the following, we will show that even in the machine scheduling framework our problem has not been studied before.

3.2. Related works in the machine scheduling literature

PAT-AS can be restated as a machine scheduling problem by taking patients as jobs, tests as job's operations (tasks), and operators as machines. These terms will be used interchangeably in the rest of the paper. In addition, we stress that the terms *task*, *test*, and *operation* will all mean the same in the following. Rooms act as renewable resources. While the room-constrained variant has mostly been disregarded in health care appointment scheduling, the issue of renewable resources has been around for a long time in the machine scheduling community (see Section 8.4 in [44]).

We focus on papers addressing makespan minimization (denoted as C_{max} in the machine scheduling jargon). Based on the features discussed in Section 2, PAT-AS belongs to the class of non-preemptive *Open Shop Problems* (OSPs) [45,46]. Recall the standard 3 fields classification (α, β, γ) introduced in [47], where α represents the machine environment, β the job characteristics, and γ the optimality criterion. Given n as the number of jobs and m as the number of machines, one per operation, in PAT-AS we have $\alpha = Om$ (an OSP with m machines) with $m = n_T$ (due to the one-operator-per-test feature), $\beta = (op \leq m)$, meaning that the number of operations in a job may be lower than m due to different test packages, and $\gamma = C_{max}$. Multiple classes, i.e., some patients do not take all tests, is known as the *missing operations* case [48].

OSP is a well-studied problem. It admits polynomial time algorithms for the 2-machine C_{max} case ($O2||C_{max}$) but turns *NP-Hard* for 3 machines unless a machine is dominant, i.e., the processing time of the shortest task on the dominant machine is no shorter than the one of the longest tasks on any other machine. In such a case, a polynomial time algorithm is provided in [49]. In addition, when the ratio of Π_{max} (the load of the bottleneck machine, that is the one with maximum load) over the longest task processing time is sufficiently large, then the problem admits a polynomial time solution algorithm yielding a makespan equal to Π_{max} [50]. In [51] the 3-machine case is proved polynomial for such a ratio larger than 7.

PAT-AS has tight links to a particular OSP, the (machine) *Proportionate* OSP (m-POSP), which is denoted by $(m - prpt)$ in the second field β [52]. In m-POSP, processing times are only machine-dependent, so that all jobs entail the same amount of processing time on the same machine, as for PAT-AS. However, m-POSP assumes $op = m$, i.e., each job is made of as many tasks as machines and jobs are then all identical, that is a PAT-AS in which all patients take the same test package. If $n \geq m$ m-POSP is optimally solved in polynomial time by the so-called *Rotation Scheduling* algorithm ($RotS$) [53], while it is *NP-Hard* for $n \geq 3$ jobs and $m > n$. In [54] a procedure similar to RotS is proposed (basically, it uses the reverse order in the cyclic sequence) which is optimal for $n > m$ and guarantees a worst-case performance ratio of $(2 - \frac{1}{n})$ otherwise. Finally, in the case of multiple operators equally skilled for the same test, the reference problem is the multi-processor m-POSP variant (MPOSP), which is dealt with in [55].

We now take a detailed look at how RotS works. First, it orders tasks and machines based on longest processing time, that is, task 1 is the one with the longest processing time, task 2 the second longest, and so on; likewise, machine m_1 is the one performing task 1, m_2 the one performing task 2, and so on. Note that m_1 is the bottleneck machine as well as the dominant one since all jobs are made of m tasks. Then, jobs are sequenced on the machines. Let us denote the n jobs as j_1,.., j_n. According to $RotS$, machine m_1 executes j_1, followed by the other jobs in lexicographical order (i.e., $j_2, j_3, \ldots, j_n$). On each other machine m_i, jobs are scheduled starting from j_i, followed by j_{i+1} and so on, with j_1 following j_n. It is easy to show that all jobs can be completed within Π_{max}, that is the workload of the bottleneck machine m_1. Indeed, while m_1 is processing j_h, machine m_i can process the $(1+((i+h-2)mod(n)))$th job, since, trivially, task i lasts no longer than task 1, whatever the jobs. To focus on jobs, now consider a generic job j_h: (i) if $m \leq h \leq n$, then j_h starts its execution on the last machine, in $(h-m+1)$th position, and then it proceeds on the other machines, from m_{m-1} to m_1, following a shortest-task-first criterion; (ii) if $h < m$, then j_h starts its execution on machine m_h, as its first job, then it proceeds on machines $m_{h-1}, m_{h-2}, \ldots, m_1$, processing tasks of increasing duration up to the longest one, then followed by machines $m_m, m_{m-1}, \ldots, m_{h+1}$ that process the remaining tasks in this order, from the shortest one onward. This ordering guarantees that when the bottleneck machine ends at time Π_{max}, all the other machines have completed their operations too. The reader may refer to Fig. 4 for the sketch of a RotS solution for $m = 3$, $n = 4$.

We now discuss the room constraint. A free room represents an additional resource (besides machines) needed to perform an operation. If binding ($n_R < n$), it affects previous complexity results [56] and may question the performance of the above-mentioned heuristics, which motivates our study. In flexible manufacturing environments, this role is often played either by human operators who supervise the machines, or by tools shared by several machines, or pallets [57,58]. We speak of *renewable resources* as these are released after usage to become available to the next user. In the most general case, each job may need a specific number of resources, and this requirement may further vary from task to task within the same job. In any case, limited resource availability imposes a maximum degree of parallelism, bounding the maximum number of machines active at the same time. In PAT-AS each job requires one resource unit, and the resource is sized by the job until its last operation has been completed. According to the three fields classification scheme, the renewable resource feature is denoted in the second field β as a triple $res(\lambda, \sigma, \rho)$, where λ is the number of resource types (1 in our case), σ the resource size (number of rooms n_R), and ρ the maximum resource requirement a task may present, i.e., $\rho = 1$ in PAT-AS. With n_R and n_T denoting the number of rooms and the number of tests, respectively, and assuming one operator per test, our problem can be labeled as $(On_T|res(1, n_R, 1), m\text{-}prpt, op \leq n_T|C_{max})$ in the machine scheduling classification, where *m-prpt* points to the proportionate property referred to machines. To our knowledge, this case has never been studied. Being the core problem of the deterministic offline scheduling process in a PAT clinic adds further motivation for its investigation and sets the stage for the study of further generalizations.

4. Solution approaches

In this section, we present two types of scheduling policies referred to as online and offline, and compare them in terms of makespan and patient waiting time. Patient waiting time is an important determinant of the perceived service quality [59]. At the same time, minimizing the makespan required to serve a given set of patients improves the utilization of staff and facilities. It also minimizes operator idle time when operators have to be available for the entire period covered by the makespan.

The online policies (Section 4.1) are implemented using a discrete-event simulation model (coded using Rockwell Arena and Visual Basic for Application) that processes patients on a first-come-first-served basis to mimic the real functioning of a PAT clinic. The start time of each test, for each patient, is not determined ex-ante. Online policies have been implemented to provide a realistic benchmark for offline ones. We consider two online policies First-come-first-served Random Arrivals (*FRA*) and First-come-first-served Slot Arrivals (*FSA*). FRA and FSA policies differ in the way patient arrivals are managed: the former belongs to the class of walk-in policies, while the latter is slot-based.

The offline policy (Section 4.2) solves a Mixed Integer Linear Programming (MILP) model that determines for each patient the arrival time at the PAT clinic and the start time of each test. The optimization model's objective function is hierarchical: Makespan C_{max} is minimized, subject to $C_{max} \leq H$ as a hard constraint, yielding C^*_{max} when feasible. Then, total patient waiting time is minimized over all feasible schedules that serve all patients within C^*_{max}. Even smoothed by the room constraint, makespan is trivially expected to improve when booking ahead of schedule, as the offline booking takes advantage of perfect information. Nevertheless, the offline policy may not be a viable option since solving the mathematical model for real-size instances may heavily degrade system performance as computing time could overly increase. In search for a trade-off between solution quality and computational burden, a (math-) heuristic approach has been explored (see Section 4.3), based on heuristically fixing some ordering decisions among those present in the model while complying with the room constraint.

4.1. Online schedulers

As pointed out in the introduction, FRA and FSA online policies manage patients' arrivals differently, while they process patients the same way once arrived. With both online policies (FRA, FSA), for each instance, the number of patients to be processed and their class are known ex-ante (and are the same as those used in offline policies) but the order in which patients arrive at the PAT is randomized, in each simulation run, by sampling without replacement from the set of patients to be processed. The two policies, however, differ in the way the patients' arrival time is determined.

FRA requires patients to arrive at the PAT clinic within Δ minutes before the clinic's closing time, where Δ is the duration of the longest test package. This is to prevent the arrival of patients too close to the closing time from resulting in too large values of operators' overtime. The arrival time is thus determined, for each patient, by multiplying $H-\Delta$ by rnd() where rnd() is a function returning a real random number in the range $[0, 1)$. That way, arrivals are spread randomly over the time window $[0, H-\Delta]$.

In the FSA policy, patients are supposed to arrive in batches of size equal to the number of rooms n_R. The batches' inter-arrival time is equal to the average service time $\bar{\delta}$. The value of $\bar{\delta}$ can be determined from empirical data as $\sum_{c\in C} \sigma_c f_c$ where f_c is the percentage of patients processed for each class c and σ_c is the total length of the tests associated with each class c. FRA and FSA allow us to compare two realistic situations. The first one assumes that patients are not given an appointment and arrive at the PAT clinic whenever they prefer, within $H-\Delta$. The second one is based on the premise that the PAT clinic cannot process more than n_R patients at a time and that executing all the tests required by a patient requires, on average, $\bar{\delta}$ minutes. Consequently, FSA allows the arrival of n_R patients every $\bar{\delta}$ minutes.

With both the online policies, for a given instance, every simulation run is thus characterized by different patients' arrival times and sequencing (for FSA the possible arrival times for the batches of patients are fixed but patients are randomly assigned to batches), and consequently different performance in terms of makespan and patient waiting time. In contrast to offline policies, with the online ones, patients can also wait outside the exam room if there are no empty rooms when they arrive. Also, with online policies, the makespan may exceed H, thus resulting in overtime.

Both with FRA and FSA policies, a patient upon arrival either seizes an available exam room or joins the queue of the patients waiting for an exam room. As soon as a room becomes available, the first patient in the queue seizes the room and releases it at the end of the last test. In case more than one room is available, the room to seize is randomly determined. Once in the room, the patient randomly seizes, among the available operators, one able to perform one of the tests needed. If no operator is available, the customer waits inside the exam room. Once a test is over, the operator who delivered the test is released, and is seized by the patient needing their skills, if any, who has been waiting for the longest in the exam room. We made this assumption to obtain a benchmark where patients are processed in the most equitable way possible. However, in real settings, the operator may not know who is waiting for the longest – unless there is a timer indicating how much time has passed since the previous operator left the room – and, consequently, will select patients randomly. If none of the patients in the exam rooms need the skills of the released operator, the operator remains idle. Both FRA and FSA, assume that the duration of each test is deterministic, that empirical data on the patient mix are known, that all patients arrive at the PAT before the closing time, and that all patients are processed, resorting to overtime when needed.

4.2. A MILP model for the offline approach

The offline booking option – denoted as OPT in the following – is the most restrictive one in terms of booking freedom as patients are all scheduled in advance. They are expected to arrive according to their given starting service time, and thus experience no waiting time outside the room. In practice, requests are collected offline and once a certain amount of service demand has been reached, expressed in terms of a given percentage (see Section 5) of the planning horizon duration H times n_R, a mathematical model is solved, which yields the appointment time of each patient. As any offline approach, OPT takes full advantage of the perfect knowledge of the demand to optimize resource exploitation and patient time as well. Moreover, the optimal solutions provide a benchmark for evaluating the schedules obtained by FRA and FSA.

Next, we introduce a network flow-inspired MILP formulation that exploits the problem structure. Note that the proposed model is valid for any resource setting different from the one analyzed in this study, i.e., it is valid for any value of n_R and n_T provided that $n_O = n_T$ and would require very minor changes otherwise. Three decision layers are present, mutually intertwined: (i) the patient set is partitioned into n_R subsets (one per room) and totally ordered within each subset; (ii) for each patient, the patient's tests are totally ordered; (iii) for each operator, the set of tests to be delivered is sequenced. Patients and their tests are modeled as nodes of the respective directed graph. Graph arcs, when selected, denote immediate precedence between two activities, thus defining the ordering decisions in the three above-mentioned layers.

Figs. 2 and 3 provide a pictorial representation of a solution for a case with 5 patients, 2 rooms, and 3 tests, color-coded as Orange (O), Blue (B) and Green (G). In detail, Fig. 2 on the left depicts two direct cycles through node 0 on the patients graph G^P, where there is

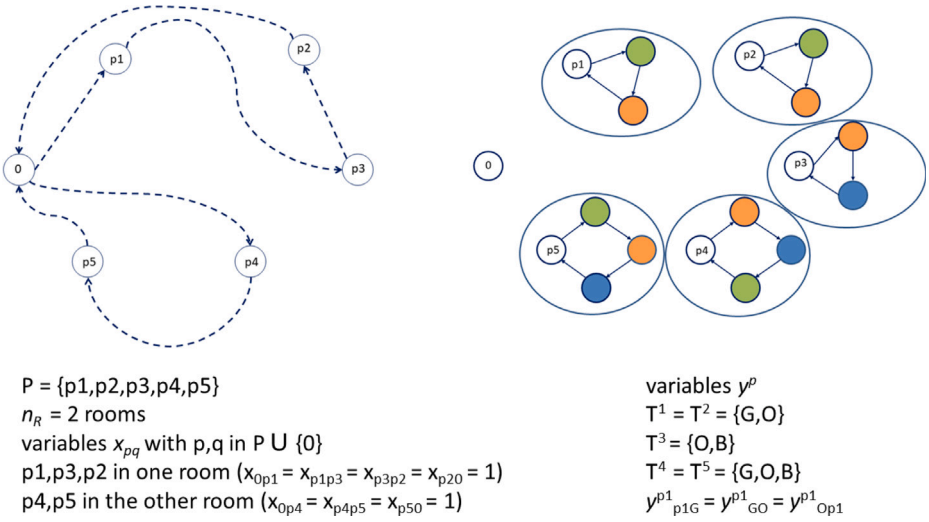

Fig. 2. On the left, a solution for the patients sequencing problem. Each tour through dummy node 0 describes the set of patients admitted into one room and their access order. Depicted arcs correspond to x variables with value 1. On the right, a feasible solution for the test sequencing problem, for each patient. Depicted arcs correspond to y variables with value 1. For example, patients p_4 and p_5 belong to the same class, requiring the Orange (O), the Blue (B) and the Green (G) test. While p_4 takes O first, then B, and finally G, patient p_5 follows the G, O, B sequence. In each test graph, the patient node p acts as the dummy node at which the cycle is closed. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

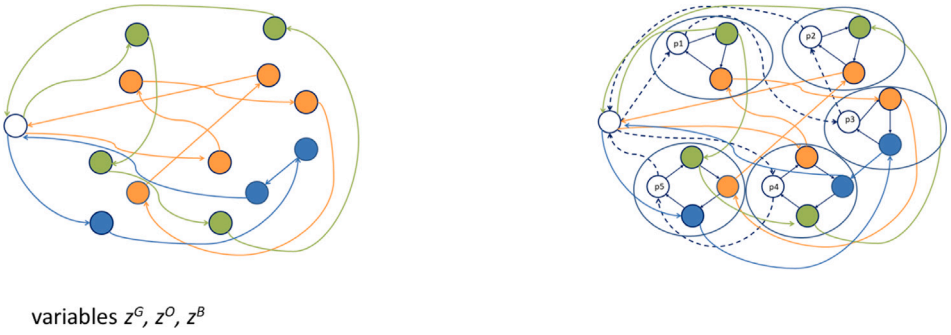

Fig. 3. On the left, a solution on the Precedence Graph for the operators sequencing problem. Depicted arcs correspond to z variables with value 1. On the right, a global picture is obtained by merging the solution of each decision layer. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

a node for each patient plus a dummy node 0. Each cycle refers to a room, i.e., patients p_1, p_3, p_2 access one room in this order while p_5 is served after p_4 in the other room. Fig. 2 on the right shows the test sequence for each patient p on graph G^{T^p}, whose nodes are the patient node p and one for each test in T^p. Consider for example patient p_4 who takes all three tests: the figure shows a hamiltonian cycle through node p_4, traversing nodes O, B, and G in this order. Tests can be done in a different order for patients with the same set of tests (p_4 and p_5 in the picture). Fig. 3 on the left depicts the operators' activity as a hamiltonian cycle for each operator o^t on graph G^t through the dummy node 0 covering the operator's tests. The node set in G^t is made of node 0 plus the node of test t for each patient p such that $t \in T^p$. Finally, Fig. 3 on the right provides a global picture of the solution: as usual, to be feasible, no direct cycles must arise in the subgraphs induced by the selected arcs (apart from those through dummy nodes). The makespan is the length of the critical path.

Table 2 summarizes decision variables: arc flow variables x, y, and z for each decision layer, and time variables τ and α for their interactions. Arcs denote immediate precedence, that is, if a generic arc (i, j) is selected in the solution it means that activity j is scheduled right after activity i. For each set of entities, the model searches for cycles through a dummy node on each precedence graph. This differs from the more popular *disjunctive* models, where arcs and variables describe a general (not necessarily immediate) precedence relation. In this respect, we argue that: (i) regarding patients, this option allows modeling rooms without introducing them explicitly, thus avoiding a potential source of symmetry; (ii) concerning a patient's tests, their number is so small that the impact of the modeling option is negligible; (iii) in reference to the tests of the same operator, the current modeling choice paves the way for a further generalization regarding multiple operators skilled for the same test.

Table 2
Decision variables of the MILP model.

$x_{pq} \in \{0,1\}$	Binary variables modeling immediate precedence between patients in the same room: if $x_{pq} = 1$ then q is the next patient to enter the room once p has left.
$y^p_{ts} \in \{0,1\}$	Binary variables modeling the test sequence of a patient such that if $y^p_{ts} = 1$ then test s is delivered right after t.
$z^t_{pq} \in \{0,1\}$	Binary variables modeling operator activity such that if $z^t_{pq} = 1$ then operator o^t attends patient q right after p.
$\tau^{in}_p, \tau^{out}_p \in R^+$	Check in and check out time for patient p.
$\alpha^p_t \in R^+$	Start time of test t for patient p.
$C_{max} \in R^+$	Makespan, where C^*_{max} denotes the minimum one.
$w_p \in R^+$	Waiting time in room for patient p.

Model (1)–(25) provides a mathematical representation of the problem.

In particular, the patient sequencing problem is modeled as an n_R-TSP on the patients graph G^P, where n_R flow units (variables x_{pq}) leave dummy node 0 (constraint (6)) while each other node is traversed

by a flow unit (7)–(8). In detail, $x_{pq} = 1$ if either patient p is scheduled immediately before patient q in the same room, or (case $p = 0$) q is the first patient, or (case $q = 0$) p is the last patient. Constraints (9) are the well-known Miller-Tucker-Zemlin inequalities (MTZ). In addition to ensuring subtour elimination, MTZ set the value of check-in (τ_p^{in}) and check-out time (τ_p^{out}) of each patient p, setting the time interval $psp(p)$ in which patient p receives service (constraints (10)).

Binary variables y_{ts}^p and constraints (11)–(14) realize a total order on the tests of each patient by searching for an hamiltonian cycle on each graph G^{T^p} (see Fig. 2 on the right). In detail, $y_{ts}^p = 1$ if patient p either takes test t right before test s, or (case $t = p$) s is the first test, or (case $s = p$) t is the last test. The start time $\alpha_t^p \in R^+$ of each test $t \in T$ is set according to the order described by y variables (14). In particular, according to (14), if t immediately precedes s, i.e., $y_{ts}^p = 1$, then its end time $\alpha_t^p + d_t$ is no greater than α_s^p, the start time of s. Moreover, constraints (15)–(16) ensure that the time interval $[\alpha_t^p, \alpha_t^p + d_t]$ devoted to test t in T^p lies within $psp(p)$. Such period is further restricted to $\left[(\tau_p^{in} + \sum_{s\in T^p} y_{st}^p d_s), (\tau_p^{out} - \sum_{s\in T^p} y_{ts}^p d_s)\right]$ in case t is neither the first test delivered to p nor the last one.

Finally, binary variables z_{pq}^t are introduced for each test t and for each pair of patients p, q such that $t \in T^p \cap T^q$, or $p = 0$ and $t \in T^q$, and vice versa: they define the activity of the operator devoted to test t (17)–(19). In detail, $z_{pq}^t = 1$ if either o^t serves p right before q, or (case $p = 0$) q is the first patient, or (case $q = 0$) p is the last patient in the operator's schedule. Time variables α_t^p must comply with such precedence constraints (20). Finally, constraints (21)–(25) define variables' domain.

The parameters M^P, M^{T^p}, and M^t in MTZ-like inequalities (9), (14) and (20) may take the default value H or be tuned more tightly depending on the instance.

The objective function (1) primarily minimizes system makespan. Assuming 0 as the starting time, the makespan of a given schedule is the maximum check out time (2) which, without loss of generality, can be assumed an integer value; as all patients must be processed within H (3), then the minimum value of C_{max} is no greater than H for any feasible instance, therefore constraint (4) is redundant. Nevertheless, explicitly setting a bound on the variables usually speeds up convergence, which motivates (4). The patient's idle time w_p, given by the time spent in the room minus service time δ^p (5), is the second term of the objective function and it is bounded from above by $Hn_R - \sum_{p\in P} \delta^p$. The weight parameter W has to be large enough to guarantee a hierarchical objective function, such as $W = Hn_R$ which is a bound for $\sum_{p\in P} w_p$.

$$min \quad W\, C_{max} + \sum_{p\in P} w_p \tag{1}$$

$$C_{max} \geq \tau_p^{out} \qquad \forall p \in P \tag{2}$$

$$\tau_p^{out} \leq H \qquad \forall p \in P \tag{3}$$

$$C_{max} \leq H \tag{4}$$

$$w_p = \tau_p^{out} - \tau_p^{in} - \delta^p \qquad \forall p \in P \tag{5}$$

$$\sum_{q\in P} x_{0q} = n_R \tag{6}$$

$$\sum_{\substack{p\in P\cup\{0\},\\ p\neq q}} x_{pq} = 1 \qquad \forall q \in P \tag{7}$$

$$\sum_{\substack{q\in P\cup\{0\},\\ q\neq p}} x_{pq} = \sum_{\substack{r\in P\cup\{0\},\\ r\neq p}} x_{rp} \qquad \forall p \in P \tag{8}$$

$$\tau_p^{out} \leq \tau_q^{in} + (1 - x_{pq})M^P \qquad \forall p, q \in P \tag{9}$$

$$\tau_p^{out} \geq \tau_p^{in} + \delta^p \qquad \forall p \in P \tag{10}$$

$$\sum_{t\in T^p} y_{pt}^p = 1 \qquad \forall p \in P \tag{11}$$

$$\sum_{s\in T^p\cup\{p\}} y_{ts}^p = 1 \qquad \forall t \in T^p, \forall p \in P \tag{12}$$

$$\sum_{s\in T^p\cup\{p\}} y_{st}^p = 1 \qquad \forall t \in T^p, \forall p \in P \tag{13}$$

$$\alpha_t^p + d_t \leq \alpha_s^p + (1 - y_{ts}^p)M^{T^p} \qquad \forall t, s \in T^p, \forall p \in P \tag{14}$$

$$\alpha_t^p \geq \tau_p^{in} + \sum_{s\in T^p} y_{st}^p d_s \qquad \forall t \in T^p, \forall p \in P \tag{15}$$

$$\alpha_t^p + d_t + \sum_{s\in T^p} y_{ts}^p d_s \leq \tau_p^{out} \qquad \forall t \in T^p, \forall p \in P \tag{16}$$

$$\sum_{p\in P(t)} z_{0p}^t = 1 \qquad \forall t \in T \tag{17}$$

$$\sum_{\substack{q\in P(t)\cup\{0\},\\ q\neq p}} z_{pq}^t = 1 \qquad \forall p \in P, t \in T^p \tag{18}$$

$$\sum_{\substack{q\in P(t)\cup\{0\},\\ q\neq p}} z_{qp}^t = 1 \qquad \forall p \in P, t \in T^p \tag{19}$$

$$\alpha_t^p + d_t \leq \alpha_t^q + (1 - z_{pq}^t)M^t \qquad \forall t \in T, \forall p, q \in P(t) \tag{20}$$

$$x_{pq} \in \{0, 1\} \qquad \forall p, q \in P \cup \{0\}, p \neq q \tag{21}$$

$$y_{ts}^p \in \{0, 1\} \qquad \forall t, s \in T^p \cup \{p\}, t \neq s, \forall p \in P \tag{22}$$

$$z_{pq}^t \in \{0, 1\} \qquad \forall t \in T^p \cap T^q, \forall p, q \in P(t) \cup \{0\}, p \neq q \tag{23}$$

$$\tau_p^{in},\ \tau_p^{out},\ w_p \geq 0 \qquad \forall p \in P \tag{24}$$

$$\alpha_t^p \geq 0 \qquad \forall t \in T^p, \forall p \in P \tag{25}$$

This network-flow model allows an implicit representation of rooms. On the contrary, explicit patient indexing introduces a level of symmetry in the model. Since all patients in the same class are identical, equivalent solutions can be obtained by simply adopting a permutation of the patient indexes within each class. This source of symmetry can be removed by imposing a total order on identical patients, such as the one based on the value of the starting time, which yields the following constraints.

$$\tau_p^{in} \leq \tau_{p+1}^{in} \qquad \forall p, p+1 \in P^c,\ \forall c \in C \tag{26}$$

i.e., for each class, the first patient in the class is the one who enters a room first, i.e., before anyone else in the same class, and so on. Note that constraints (26) do not affect optimality, that is, any solution is still possible.

A feasible solution sets the precedence among patients and among tasks in such a way that, for any non-null time interval I in which $n(I)$ patients have been attended (their psp intersects I) and $n(I) > n_R$, then at least $n(I) - n_R$ patients end within I. Moreover, by way of a standard transformation of the graph that inserts dummy nodes representing start and end of service (instead of a single dummy node), the subgraph induced by the solution is acyclic. On that graph, the makespan corresponds to the duration of the critical path (the longest one).

As experimentally proven in Section 5 and despite symmetry breaking, solving this model becomes computationally demanding once instances reach real-life sizes. Therefore, we propose a matheuristic that fixes some arc decision variables which are likely to be part of the critical path and let the solver build the rest of the solution around this backbone, as presented below.

4.3. A family of matheuristics for OPT

To further motivate our study, in this section, first, we claim that the performance of RotS may deteriorate because of the room constraint; second, we introduce a family of matheuristics inspired by the concepts of core decisions and bottleneck operator. The rest of the section is structured to answer the three research questions — highlighted in bold, that led to the definition of the heuristics.

Does the room constraint affect the performance of RotS? Note that POSP is solved exactly by procedure RotS, achieving $C_{max} = \Pi_{max}$,

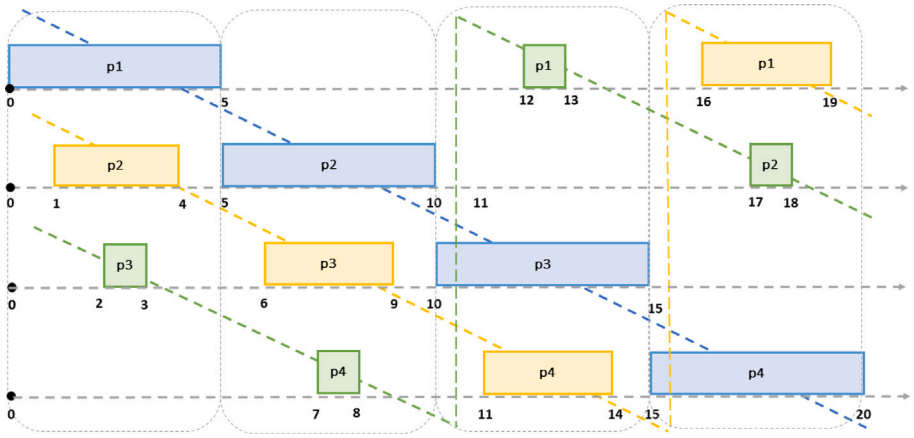

Fig. 4. One of the schedules yielded by RotS. Horizontal dark gray dotted lines are the patient timelines, colored dotted lines represent the sequence of each operator's activities. The time interval $[0, C_{max}]$ is partitioned into 4 sub-intervals, depicted as gray rectangles, during which each operator delivers one test. Many other schedules with the same operator and patient ordering and such that $C_{max} = \Pi_{max}$ can be built in order to reduce patient waiting time. However, even infringing the one-test-per-sub-interval property, total patient waiting time between tests remains significant. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

when jobs outnumber machines. Therefore, the first question concerns whether the room constraint may affect the performance of RotS. In this respect, we show that $C_{max} = \Pi_{max}$ may no longer be guaranteed.

Lemma 4.1. *If the rotation algorithm RotS must comply with the room constraint, the makespan of the provided schedule can be greater than the workload of the bottleneck operator* $\Pi_{max} = \max_{t \in T}\{|P(t)|d_t\}$ *if* $n > m$ *and* $m = n_R$ *(i.e.,* $n_P > n_T$ *and* $n_T = n_O = n_R$*).*

Proof. Let j_i denote the ith job. Note that RotS would schedule operation i of j_i as the first one to be processed on machine i, for each $i = 1, \dots, m$. In our setting ($n_R = m$) each such event requires the occupation of one free room. Therefore, when j_m is started on machine m, the last room available is occupied. Therefore, j_{m+1} can be started only once at least one job among $\{j_1, \dots, j_m\}$ has finished. Since j_i $\forall i \in 1, \dots, m-1$ is the last job to be processed on machine $i+1$, any such job cannot finish before all the other jobs have started. On the other hand, all predecessors of j_m on any machine belong to $\{j_1, \dots, j_{m-1}\}$, i.e., jobs that have an assigned room and can be processed. Therefore, j_m is the first job in $\{j_1, \dots, j_m\}$ to release a room, and j_{m+1} will occupy it. From there on, the maximum parallelism allowed is 1 (meaning that all but one job are on hold), until operation m of job n is processed, thus triggering operation m of job 1. It follows that m_1 may have idle times and, therefore, the duration of the critical path is larger than Π_{max}. □

Next, we provide a toy example that materializes the previous demonstration by showing that any schedule complying with the ordering provided by RotS and completing all jobs within the lower bound Π_{max}, cannot comply with the room constraint in case of n_T rooms for $n_T < n_P$. Consider the following: $n_P = 4$ patients, $n_T = 3$ tests, namely t_1, t_2, t_3 whose duration is $d_1 = 5$, $d_2 = 3$, $d_3 = 1$, respectively, so that $\Pi_{max} = n_P d_1 = 20$. According to RotS, operator o^1 would serve p_1, p_2, p_3, p_4, o^2 would serve p_2, p_3, p_4, p_1, and o^3 would serve p_3, p_4, p_1, p_2, each in that order, respectively. A graphical representation of a schedule complying with this ordering is provided in Fig. 4, with the following color coding: t_1 is depicted in blu, t_2 yellow, and t_3 green. Starting times of bottleneck operation t_1 are set to achieve $C_{max} = \Pi_{max}$, i.e., they take value in the set $\{kd_1, k = 0, \dots, n_P - 1\}$. This partitions the time interval $[0, C_{max}]$ into n_P sub-intervals, each one corresponding to the execution of test t_1 for one patient.

Actually, RotS yields a family of solutions in which the job that is executed as the hth on a machine other than the bottleneck one, can be processed any time within the time interval $[(h-1)d_{max}, hd_{max}]$, where d_{max} is the duration of the bottleneck operation (d_1 in our example). For example, the solution in Fig. 4 schedules the tasks associated with tests t_2 and t_3 right in the middle of such intervals, but many other choices are possible. While the sequence of operations on the bottleneck machine has no idle times – it corresponds to the critical path whose length is the makespan – there is some degree of freedom concerning the tasks on the other paths, which can be shifted backward or forward to compact some of them (such as all the tests of the same patient) within a shorter interval. As long as the orderings resulting from RotS are respected, all these solutions provide schedules that end at Π_{max} and are optimal as well.

In any schedule complying with RotS, t_1 is the first test delivered to p_1 and the last one delivered to p_4. Now, let us try to fit the ordering yielded by RotS (depicted as dotted lines in Fig. 4) into an environment characterized by 3 rooms (at least two patients use the same room) and discuss if the room constraint can be satisfied while keeping $C_{max} = \Pi_{max}$. We will show that there is no feasible schedule for any such option, namely, a schedule which complies with the previous requirements, and such that the last operation of the first patient in a room precedes the first of the second patient in the same room. In particular, (i) either a pair, say p_i and p_j, that is served by o^1 in this order, is served by another operator in the opposite order, meaning that the *patient service periods* of the two patients would intersect, and they cannot be feasibly served in the same room, or (ii) the makespan would increase. This can be verified in Fig. 4. Let us consider the pairs that operator o^1 (blue) serves in sequence: for the pair (p_1, p_2), note that o^2 (yellow) serves p_2 as first and p_1 as last; for the pair (p_2, p_3), operator o^3 (green) serves p_3 as first and p_2 as last; moreover, p_2 could not use the same room of p_1 since p_2 receives test t_2 while p_1 is receiving test t_1. It follows that p_1, p_2 and p_3 need one room each. Therefore, p_4 must use a room that was previously occupied by another patient. Recall that p_4 must receive t_1 at time 15 and the ordering of the tests is t_3, t_2, t_1. Therefore, p_4 cannot use p_3's room. However, p_4 cannot use either p_1's room or p_2's room in a feasible way, since o^3 serves p_4 before p_1 as well as before p_2.

Fig. 5 provides a pictorial representation of a solution fulfilling the ordering of RotS and attaining $C_{max} = \Pi_{max}$, where p_1 and p_4 are both served in room 1, and the starting times of t_2 and t_3 have been adjusted so that the total patient waiting time is minimum. Specifically, $psp(p_1)$ = (0,14), $psp(p_2)$ = (2,11), $psp(p_3)$ = (4,15), and $psp(p_4)$ = (6,20). As graphically depicted, $psp(p_1)$ and $psp(p_4)$ are not disjoint.

Do makespan and workload of bottleneck operator coincide when the room constraint is imposed? To show that it does not, suppose that the longest test does not belong to all the patient packages. It is easy to build instances where the optimal makespan C_{max} is strictly greater than the workload Π_{max} of the bottleneck operator, i.e., the busiest one. As an example, consider the case where $n_P = 6$ patients,

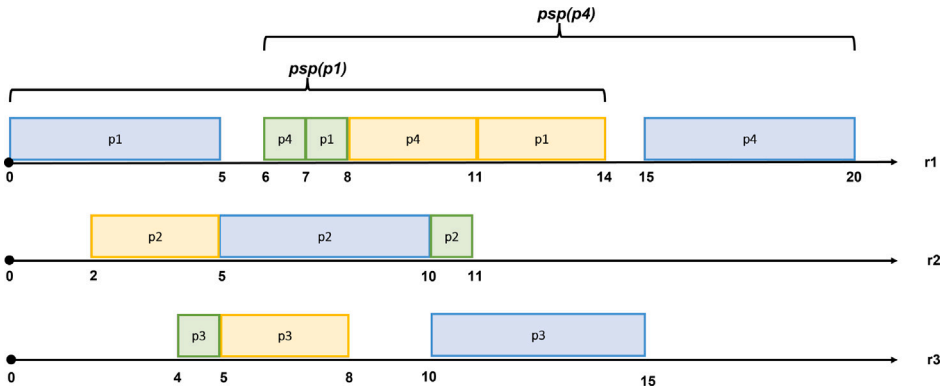

Fig. 5. No schedule complying with RotS and the room constraint may attain $C_{max} = \Pi_{max}$ in a n_T rooms environment if $n_T < n$.

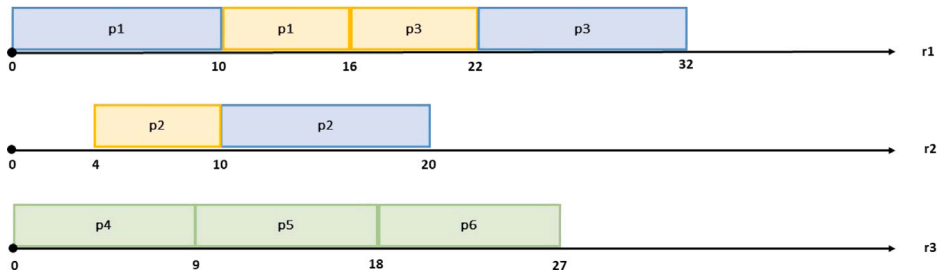

Fig. 6. No schedule with $C_{max} = \Pi_{max}$ does exist.

$n_T = 3$ tests, namely t_1, t_2, t_3 whose duration is $d_1 = 10$, $d_2 = 9$, $d_3 = 6$, respectively. In addition, suppose that 3 patients must undergo a package consisting of tests t_1 and t_3, while the other 3 patients must undergo only test t_2. The bottleneck operator is the operator in charge of test t_1 and $\Pi_{max} = 30$. The optimal makespan for this instance is 32, strictly greater than $\Pi_{max} = 30$, as Fig. 6 shows.

Is bottleneck operator still crucial when the room constraint is imposed? Now it is clear that RotS cannot solve our problem because of the room constraint. Nevertheless, RotS may inspire the search for a restricted set of precedence relations to be fixed that provide an efficient trade-off between the computational effort needed for solving the remaining subproblem and the solution quality degradation.

The procedure is made of the following steps:

- let us define the bottleneck operator as the operator with the highest workload. More formally,

  $$bo = \arg\max_{t \in T} \{|P(t)| \cdot d_t\}$$

  As observed, scheduling decisions regarding the bottleneck operator identify core decisions.
- Patients requiring the test associated with *bo* are ordered according to some criterion.
- Scheduling variables $z^{bo}_{p,p+1}$ are fixed to one, where p is each of the patients requesting service from *bo* and $p+1$ is the next in the order, if any.
- Model (1)–(26) is run.

Three ordering criteria have been proposed, all based on the class a patient belongs to. Only the classes containing the test performed by the bottleneck operator are considered in the ordering. Note that patients have been given a lexicographical order within each class to break symmetry. According to the first criterion, denoted as SPT (Shortest Processing Time), the bottleneck operator *bo* serves all patients within the same class according to the lexicographical order. Then, classes are ordered according to non-decreasing service time σ^c and the last patient in one class is served by *bo* before the first one in the next class. The second criterion, denoted as LPT (Longest Processing Time), adopts the opposite ordering of the classes, starting from the one with the longest service time. Within each class, again, *bo* operates according to lexicographical order. Finally, the third criterion orders classes by lexicographical order, which is unrelated to σ^c. In the first round, *bo* serves the first patient (in lexicographical order) in class one, followed by the first one (in lexicographical order) in class two, and so on until the last (in lexicographical order) class. In the next round, the procedure repeats involving the second patient (in lexicographical order) of each class, and so on until all patients that must undergo the test of *bo* have been processed. This last criterion is denoted as RR (Round Robin).

Note that nothing has been said about how tests are sequenced within each set T^p, nor on the side of the other operators. Nevertheless, as we report next, the subproblem that is obtained by adding the ordering decisions can be quickly solved. As the result is not provably optimum these approaches are matheuristics.

5. Experimental results

In this section, we present and discuss the computational results obtained by the abovementioned approaches on a plurality of scenarios. Specifically, Section 5.1 describes the real instances and the seven scenarios which are used to assess the applicability of the proposed approaches to the real case as well as to more general settings. Section 5.2 describes the computing environment and the KPIs used to evaluate solution quality and computational efficiency. Then, Section 5.3 reports the results obtained on the first three scenarios which are related respectively (1) to the real case, (2) to the implications of splitting the working day in two sessions, and (3) to some extensions concerning the relationships between test duration. Section 5.4 further explores the latter issue: we extend the analysis to another four scenarios and drop the assumption that each package includes the longest test. This aims to generalize the results to different settings and to support the claim that findings do not depend on the real case particular structure. A further step in the generalization of the results is taken in the direction identified by the counterexample shown in Section 4.3 in

Table 3
Experimental study in brief. Instances are characterized by: $n_R = 4$ rooms, $n_T = 4$ tests, $n_O = 4$ not-cross trained operators, one for each test; $|C| = 4$ patient classes, with $C = \{C_1, C_2, C_3, C_4\}$; opening time $H = 4$ or 8 h; $T = \{t_1, t_2, t_3, t_4\}$, with d_i duration of test t_i; t_1 is the longest test. Patient class C_i is defined in terms of tests t_j: if patients in class i need t_j, an 1 is reported in columns Patient classes, 0 otherwise. σ^c is the total processing time for patients in class c, i.e., the sum of needed test times for a patient class.

Scenario	Assumption	Objective of the study	H (in min)	Test duration				Patient classes						Number of instances
				d1	d2	d3	d4	Class	t1	t2	t3	t4	σ^c	
S1	Longest test is in every patient class $d1 > (d2 + d3 + d4)$	Impact of patient mix on computational results	480	28	9	6	7	C_1	1	1	0	0	37	Generated exhaustively all possible patient mixes, with (i) at least one patient in each class, (ii) service time of at least 40% of total available time $H \cdot n_R$, (iii) workload of bottleneck operator not exceeding H. Instances considered = 98
								C_2	1	1	1	0	43	
								C_3	1	1	0	1	44	
								C_4	1	1	1	1	50	
S2	Same as above	Impact of having two 4 h session.	240	Same as above				Same as above						Same as above Instances considered = 69
S3	Longest test is in every patient class $d1 \leq (d2 + d3 + d4)$	Impact of altering test durations. Halve d1	240	14	9	6	7	C_1	Same as above				23	Same as above instances considered = 1742
								C_2					29	
								C_3					30	
								C4					36	
S4	Remove t1 from patient class 1	Impact of altering class structure	240	14	9	6	7	C_1	0	1	0	0	9	Generated exhaustively all possible patient mixes, with (i) at least one patient in each class, (ii) service time of at least 45% of $H \cdot n_R$, (iii) workload of bottleneck operator not exceeding 75% of H. Randomly selected 100 instances.
								C_2	1	1	1	0	29	
								C_3	1	1	0	1	30	
								C_4	1	1	1	1	36	
S5	Remove t1 from patient class 2	Impact of altering class structure	240	14	9	6	7	C_1	1	1	0	0	23	Randomly selected 100 instances.
								C_2	0	1	1	0	15	
								C_3	1	1	0	1	30	
								C_4	1	1	1	1	36	
S6	Remove t1 from patient class 3	Impact of altering class structure	240	14	9	6	7	C_1	1	1	0	0	23	Randomly selected 100 instances.
								C_2	1	1	1	0	29	
								C_3	0	1	0	1	16	
								C_4	1	1	1	1	36	
S7	Remove t1 from patient class 4	Impact of altering class structure	240	14	9	6	7	C_1	1	1	0	0	23	Randomly selected 100 instances.
								C_2	1	1	1	0	29	
								C_3	1	1	0	1	30	
								C_4	0	1	1	1	24	

which the bottleneck operator is not present in all test packages. We then generated an additional set of instances with these characteristics whose results are evaluated in Section 5.5. In this set of instances, we also show the effect of using a different objective function that takes into account the idle time of the operators. Finally, Section 5.6 takes a global view of the results to provide a comparison among scenarios and suggest possible takeaways.

5.1. Instance and scenario description

An instance is characterized by the following input variables: number of rooms n_R, number of tests n_T, number of operators n_O, test duration, time horizon H, patient classes, patient mix. Real instances are characterized by: (i) 4 rooms ($n_R = 4$), (ii) 4 tests ($n_T = 4$), (iii) 4 operators ($n_O = 4$), (iv) 4 patient classes ($|C| = 4$), (v) an opening time of 8 h ($H = 480$). In this study, we assume that each operator masters a different skill.

Patients' mix, i.e., the percentage of patients in each class can be retrieved from the analysis of the empirical data at the clinic. Based on these data, the probability that a patient belongs to a certain class, considering classes ordered on ascending service time, i.e., $\sigma_c < \sigma_{c+1}$, is the following: 31% for class C_1, 14% for class C_2, 16% for class C_3, and 39% for class C_4. More details on the real case study that inspired this work are given in Appendix A.

The real scenario is referred to as S1. Further generalizations are obtained by varying the opening time H (scenario S2), the length d_1 of the longest test (scenario S3), and the structure of the test classes (scenario S4–S7). Specifically, Table 3 reports the assumptions characterizing each scenario, the objective of the study, H, test duration, the composition of classes in terms of tests, service time of patient classes, number of instances considered, and criteria used to generate them.

Table 4
KPIs: names and description.

KPI name	KPI description
MKS	Makespan
%BO	Percentage difference between makespan and workload of the bottleneck operator
WTI	Total waiting time inside exam room
%GAP	percentage gap
Time	Computational time

5.2. KPIs, policies and computing environment

Key Performance Indicators (KPIs) refer to solution quality and efficiency. With respect to the quality of the solution, we consider: (i) makespan, (ii) the percentage difference between makespan and workload of the bottleneck operator, and (iii) total patient waiting time inside the exam room. With respect to the computational efficiency of the offline approaches, we consider (i) solution time, and (ii) the percentage gap between the best solution found and the best lower bound returned by the solver. The name and description of KPIs are listed in Table 4 and they will be used hereafter.

Both online and offline policies are tested and compared in terms of the KPIs described above. Specifically, we test 4 offline approaches (OPT and the three variants of the matheuristic: SPT, LPT, and RR) and 2 online ones (FSA and FRA). Not all policies have been used in all scenarios and not all KPIs are applicable to all policies: in the following sections, scenario by scenario we detail the computational tests done.

The numerical analysis has been performed on a PC equipped with an AMD® Ryzen 9 3950x 16-core processor x32 and 32 Gb of RAM. The optimization model and the matheuristics have been coded in C++ and solved using the IBM ILOG Cplex 12.10 solver imposing a time limit

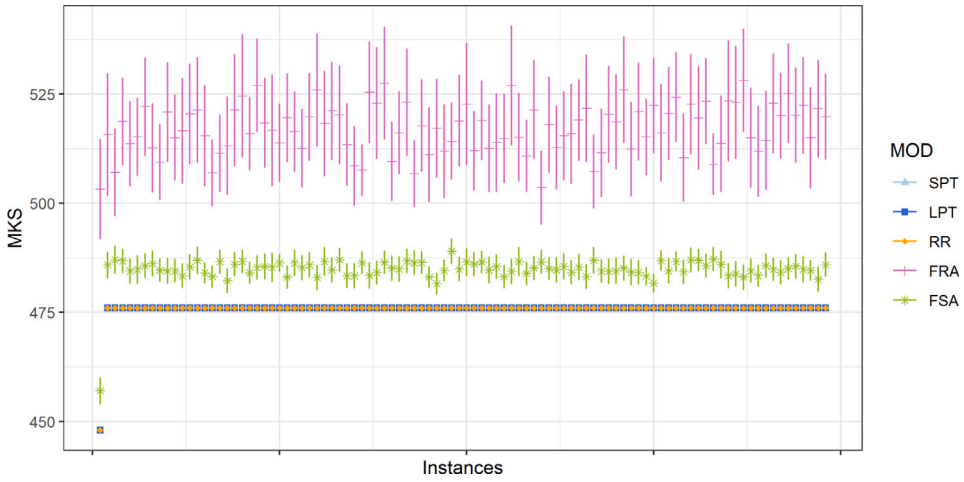

Fig. 7. Scenario S1, $H = 480$, $d_1 = 28$: Makespan (in minutes).

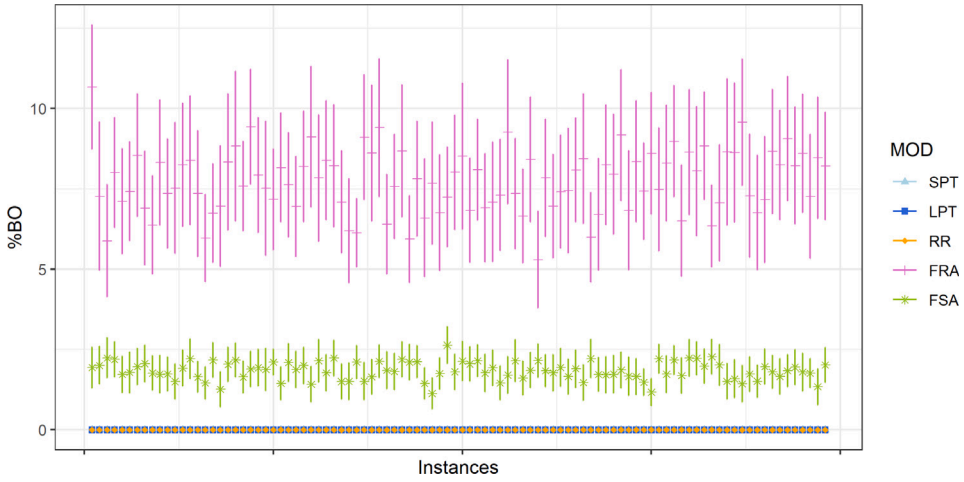

Fig. 8. Scenario S1, $H = 480$, $d_1 = 28$: percentage gap between the workload of the bottleneck operator and makespan.

equal to 3600 s and a memory limit for the branch and bound tree equal to 8 Gb.

5.3. Computational results for scenarios S1–S3

In scenario S1, characterized by $d_1 = 28$ and $H = 480$, the exhaustive generation of all possible patient mixes, with at least one patient in each class, a service time at least 40% of the total available time, and the workload of the bottleneck operator not exceeding H, resulted in 98 instances. While OPT fails on all instances, being unable in all cases to find a feasible solution to model (1)–(26) within the given time and memory limits, all matheuristics SPT, LPT, and RR provide for each instance the optimal makespan and a very low waiting time inside the exam room (below 33 min for all instances).

Figs. 7 and 8 show, for each instance (x-axis) and for all approaches, the makespan MKS and the percentage difference between the load of the bottleneck operator and the makespan (%BO), respectively. For FRA and FSA the plot shows the 95% confidence intervals for the mean obtained with 30 simulation runs. The figures clearly show that: (i) instances are challenging being the makespan very close to opening time H, (ii) the matheuristics always return an optimal solution with respect to the makespan; in fact, the makespan is the same as the load of the bottleneck operator, (iii) for FRA and FSA, makespan always exceeds the opening time and overtime can be very high, especially for FRA, (iv) across instances, FRA performs significantly worse than FSA.

Fig. 9 reports the total waiting time, i.e., the sum (over all patients processed) of WTI for SPT, LPT and RR and the 95% confidence intervals for the mean for FRA and FSA. We can observe that: (i) all the matheuristics return, on average, significantly better results than the online approaches. In the worst case scenario, the maximum WTI for the offline approaches is equal to 47 min while for the online approaches, the maximum WTI is equal to 1034.1 and 1049.8 min for FRA and FSA, respectively (for the online approaches the max(WTI) is calculated as the maximum of individual replications maxima), (ii) contrary to what happens for the overtime, FRA is significantly better than FSA in terms of WTI, (iii) the average waiting time per patient inside the exam room across all instances is 1.8, 1.9 and 1.9 min respectively for SPT, LPT and RR, and to 40.48 and 57.9 min respectively for FRA and FSA.

In terms of computational efficiency, all instances are solved to optimality by all the matheuristics. For SPT, LPT and RR the mean value of the computational time is equal to 139.6, 180.5, 158.4 s, respectively, while the maximum value is 735.8, 1468.5 and 784.5 s, respectively.

In scenario S2, characterized by $d_1 = 28$ and $H = 240$, the exhaustive generation of all possible patient mixes resulted in 69 instances. In this case, both OPT and the matheuristics can find a feasible solution for all instances. Fig. 10 reports %BO (for FRA and FSA the plot shows the 95% confidence intervals for the mean obtained with 30 simulation runs) and clearly shows that all of the offline approaches are always able to provide the optimal makespan (%BO = 0). Analysis

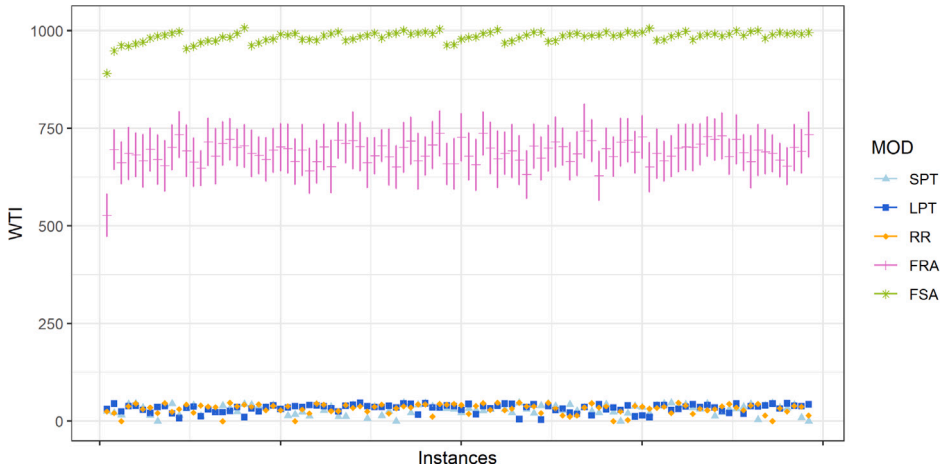

Fig. 9. Scenario S1, $H = 480$, $d_1 = 28$: Waiting time inside exam room (in minutes).

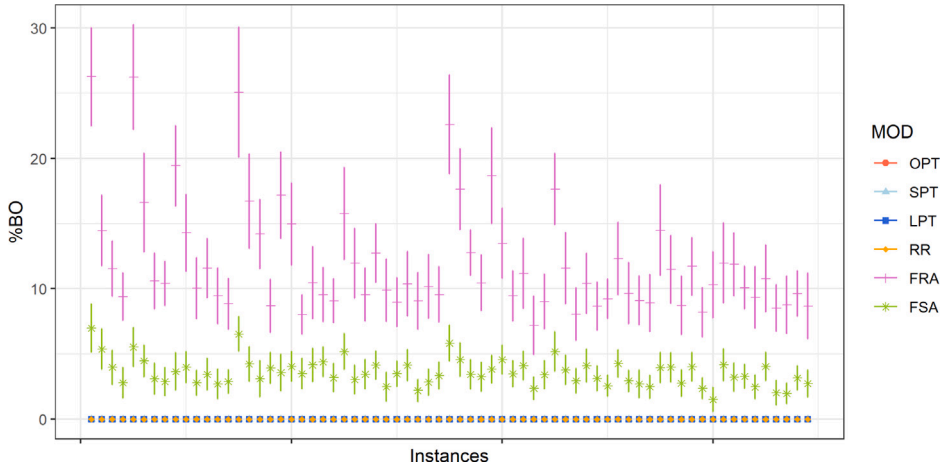

Fig. 10. Scenario S2, $H = 240$, $d_1 = 28$: percentage gap between the workload of the bottleneck operator and makespan.

of results allows the following findings: (i) the performance of OPT and of the matheuristics are very similar, in fact (ii) both provide the optimal makespan and a very low total waiting time. Like scenario S1, (iii) offline approaches dominate online ones with respect to all the solution quality KPIs, (iv) both FRA and FSA are characterized by overtime (which is significantly higher for FRA than FSA), and by (v) a substantial waiting time (which is significantly higher for FSA than FRA). Specifically, the maximum WTI is 435.8 and 472 min for FRA and FSA, respectively, 19 min for OPT and 22 min for SPT, LPT and RR; the maximum %BO is equal to 33.6% and 6.5% for FRA and FSA and 0 for the offline approaches; the average waiting times inside the exam room per patient are 24.1 (FRA), 49 (FSA), 0.8 (OPT), 0.5 (SPT), 0.9 (LPT), and 0.7 (RR) minutes.

In terms of computational efficiency, all instances are solved to optimality. For OPT, SPT, LPT, and RR the mean computational time is 19.1, 5.2, 4.4, and 4.8 s, respectively, while the maximum computational time is 179.8, 18.7, 14.5, and 16.3 s, respectively.

In scenario S3, characterized by $d_1 = 14$ and $H = 240$, the exhaustive generation of all possible patient mixes resulted in 1742 instances. A preliminary analysis of the results obtained has made it possible to observe that, similarly to what happens in the S1 and S2 scenarios, offline approaches perform better than online ones in terms of all the KPIs used.

Computational results clearly show that, on the solved instances, the 4 offline approaches are able to find the optimum or very close to the optimum in terms of makespan. For OPT, the average percentage difference %BO computed on the solved instances, is 0.1%. Only in one instance, the %BO is as high as 12.1%. For the matheuristics, the optimal makespan is achieved on 100% of the instances. We also report that for FRA and FSA, the maximum %BO is equal to 33.4% and 6.5% respectively.

Interestingly, also the average waiting time inside the exam room is very small for the offline approaches: equal to 3.7 min for OPT and 1.1 for SPT, LPT and RR. For the online approaches, instead, the average waiting time inside the exam room is 14.8 and 23.1 min for FRA and FSA respectively. The maximum waiting time inside the exam room is 505.7, 500.5, 295, 23, 30, 23 min for FRA, FSA, OPT, SPT, LPT, RR respectively. Figs. B.17 and B.18 in Appendix B provide further details.

In terms of computational efficiency, OPT finds a feasible solution for 43.2% of the instances and finds an optimal solution 4.9% of the time. The average Gap for the instances that are not solved to optimality is very small (0.1%). SPT, RR and LPT are able to find optimal solutions for all the instances.

5.4. Computational results for scenarios S4–S7

For each scenario S4–S7, we generated all the patient mixes that satisfy the following two conditions: (i) the total service time is at least 45% of the total available time ($H \cdot N_R$), and (ii) the workload of the bottleneck operator does not exceed 75% of the regular time H. H is fixed to 240 and $d_1 = 14$. Even with these assumptions, the number of instances resulting in each scenario can be high, thus we randomly selected 100 instances for each scenario, summing up to 400 additional

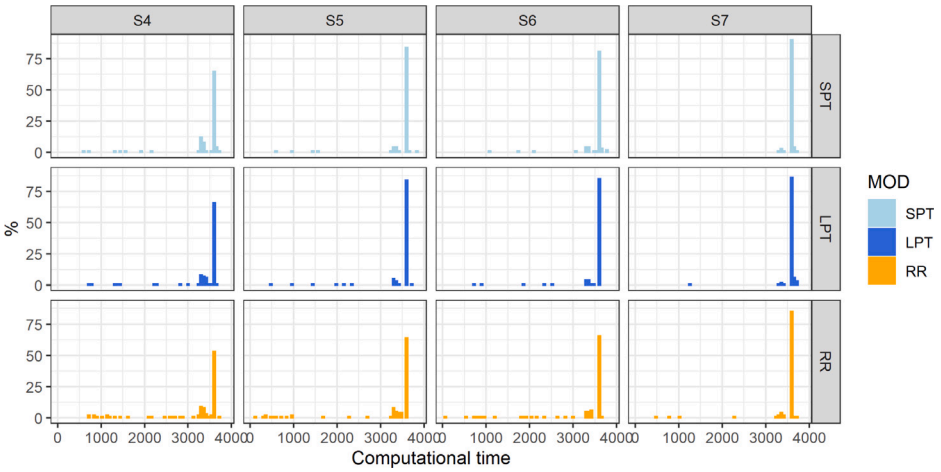

Fig. 11. Scenarios S4–S7, $H = 240$: distribution of computational time (in seconds).

Table 5
Scenarios S4-S7: an overview of solution quality. In regard to waiting time, expressed in minutes (m), the average waiting time per patient (AvgInd) and the average total waiting time (AvgTot) are reported together with the maximum total waiting time (MaxTot) computed across scenarios. In regard to the bottleneck operator, expressed in percentage (%), average and maximum values (across scenarios) of %BO are given. Average values are computed across the scenarios.

Model	Waiting time			%BO	
	AvgInd (m)	AvgTot (m)	MaxTot (m)	Avg (%)	Max (%)
OPT	3.83	67.90	186.00	2.14	12.04
SPT	3.15	57.89	145.00	5.51	15.58
LPT	3.34	61.61	160.00	5.32	11.93
RR	1.67	30.32	91.00	0.16	3.51

instances. Note that in these scenarios, the bottleneck operator is no longer the operator administering test t_1, since the maximum operator workload depends on the test duration and on the number of patients requiring that service. In the following, we present computational results relevant to the comparison among offline policies, namely, OPT and the three variants of the matheuristic, i.e., SPT, LPT and RR.

With respect to solution quality, interestingly, for most of the scenarios, there is at least one policy providing an optimal solution in terms of makespan. Scenarios S4–S7 seem to be progressively more challenging, with S4 representing the easiest instances in the test set while S7 is the most difficult. Figs. B.19–B.21 in Appendix B provide further details. An overview of KPIs relevant to solution quality is given in Table 5 where for each approach (row) summary information is given. Specifically, in regard to waiting time, the average waiting time per patient and the average total waiting time are reported together with the maximum total waiting time computed across scenarios. For what concerns the bottleneck operator, average and maximum values (across scenarios) of the percentage difference between the workload of the bottleneck operator and the makespan are given. Average values are computed across the scenarios. Overall, RR stands out as the best offline policy, while SPT and LPT seem to be quite equivalent.

Interestingly, the ranking (in terms of objective function value) among the approaches presented in Table 6 reveals that across scenarios, on average, RR ranks first or second in 91% of the instances, thus proving to be the most effective approach. The other three approaches, namely, OPT, SPT and LPT rank (on average) first or second respectively in 28%, 43%, and 43% of the cases. So, SPT and LPT seem candidates to be the second-best approach.

Finally, with respect to efficiency, Table 6 reports for each scenario and for each approach, the percentage of instances for which a feasible solution has been found (%sol), the percentage of instances for which an optimal solution has been found (%opt), the average computational time, the average gap and the maximum gap across the solved instances. Across the scenarios, the average gaps for the four approaches OPT, SPT, LPT and RR are respectively 2.18, 5.53, 5.34, and 0.17, thus confirming that RR is the best option not only in terms of solution quality but also in terms of efficiency.

Table 6 also shows, separately for each scenario (rows) and for each policy (columns), the gap distribution, i.e., the percentage of instances for which the optimality gap is within a given range (less than 1%, less than 5%, less than 10%, over 10%). RR proves to be a very effective tool since, even for the worst performing scenario (S7), in 88% of cases it finds solutions with an optimality gap lower than 1% and never returns a gap higher than 10%.

As a further analysis of the results in terms of efficiency, Fig. 11 shows that for each scenario and for each approach there is a meaningful number of instances for which the time limit (3600 s) is reached. Remarkably, a further set of computational tests run with a shorter time limit (600 s) shows that almost the same solution quality can be obtained with the worst deterioration of the gap equal to 24.63% and on average 1.84%.

5.5. The longest test is not in every patient class: further computational results

The results presented in the previous sections showed that the optimal makespan coincides with the workload of the bottleneck operator, but the counterexample provided in Section 4.3 shows that instances can be constructed for which this is no longer true. In this section, we want to further investigate this behavior experimentally. For this purpose, we generated an additional set of instances with the structure of the counterexample.

The purpose of the section is twofold: (i) to show the impact of the absence of the longest test on efficiency and solution quality; (ii) to show the effect of a different hierarchical objective function that first minimizes the makespan and second minimizes the idle time of the operators instead of patient waiting time. In the real case that inspired this study, operators are always available for the entire workday and their idle time is therefore given by the difference between the clinic's opening time and the workload resulting from the patients they must serve. Therefore, in this case, minimizing idle time loses its original meaning whereas targeting the minimum makespan indirectly reduces their idle time. Moreover, [39] reveals that, in real practice, practitioners do not see idle time as a problem since it is used for substitute tasks, such as e-consulting. Nevertheless, there may be different organizational models in which operators' shifts start when they serve their first patient and end once their last one has been served: in such a case, minimizing operators' idle time becomes crucial, potentially to the detriment of

Table 6
For each scenario and each model, columns Ranking report the approach ranking in terms of the objective function value. Then, efficiency information is reported: the percentage of instances for which a feasible (%sol) or optimal (%opt) solution is found, the average computational time (Mean Time), the average gap (Mean %Gap) and the maximum gap (Max %Gap) across the solved instances. Finally, information on gap distribution is given, i.e., the percentage of instances for which the optimality gap is within a given range.

Sc	NInst	Model	Ranking				%sol	%opt	Mean	Mean	Max	GapDistribution			
			1st	2nd	3rd	4th			time	%Gap	%Gap	<1%	<5%	<10%	≥10%
S4	100	OPT	4	29	2	44	79	1	3598	1.3	7.6	63.3	93.7	100.0	0.0
		SPT	15	28	40	16	99	30	3382	4.4	13.6	52.5	52.5	92.9	7.1
		LPT	16	24	44	16	100	33	3375	4.3	9.6	52.0	52.0	100.0	0.0
		RR	75	18	6	1	100	46	3145	0.0	0.6	100.0	100.0	100.0	0.0
S5	100	OPT	1	19	1	28	49	0	3600	1.9	18.6	59.2	89.8	95.9	4.1
		SPT	13	32	48	7	100	14	3475	5.5	14.7	41.0	45.0	68.0	32.0
		LPT	16	34	37	13	100	15	3454	5.2	11.9	46.0	49.0	70.0	30.0
		RR	75	16	9	0	100	36	3171	0.2	3.5	94.0	100.0	100.0	0.0
S6	100	OPT	1	22	3	29	55	0	3600	2.6	9.5	36.4	80.0	100.0	0.0
		SPT	13	35	47	4	99	14	3512	5.8	17.8	38.4	46.5	71.7	28.3
		LPT	16	29	36	19	100	15	3478	5.6	11.9	37.0	45.0	74.0	26.0
		RR	76	10	11	2	99	33	3211	0.1	4.5	94.9	100.0	100.0	0.0
S7	100	OPT	1	36	2	22	61	0	3600	2.9	12.5	39.3	78.7	93.4	6.6
		SPT	6	31	47	16	100	5	3588	6.4	16.2	26.0	42.0	64.0	36.0
		LPT	14	22	41	23	100	5	3570	6.3	14.5	30.0	45.0	65.0	35.0
		RR	79	13	8	0	100	13	3479	0.4	5.5	88.0	99.0	100.0	0.0

patients' waiting time. As an example, think of those contexts in which operators carry out activities both for the clinic and outside the clinic, and their coordination can therefore enable greater system efficiency. Defining an objective function in these contexts can be a difficult task since it may be crucial to make sure that idle time is compacted as much as possible, rather than broken up into short fragments of time that cannot be used for other activities, and this is not captured by just the shift duration; moreover, there may be different priorities among operators depending on the tests they perform; in addition, fairness issues may arise when, compacting an operator's shift over a short period of time can spread the activities of another operator over a much larger period. Defining such objective functions is beyond the scope of this paper, which is instead focused on patient satisfaction, so the objective function used to obtain the results in this section is simply the sum of the idle times of the operators, each computed as the difference between the shift duration and the total service time.

The additional set of instances used for these experiments are characterized by the following input variables: $n_R = n_T = n_O = 3$; an opening time of 4 h ($H = 240$); test durations $d_1 = 10$, $d_2 = 9$, and $d_3 = 6$; 2 patient classes ($|C| = 2$) with $C_1 = \{t_1, t_3\}$ and $C_2 = \{t_2\}$. As for the previous scenarios, we exhaustively generated all possible patient mixes, with at least one patient in each class, a service time of at least 40% of the total available time, and the workload of the bottleneck operator not exceeding H, resulting in 169 instances. Observe that, according to the number of patients generated in each class, the bottleneck operator can be either the one in charge of t_1 or the one in charge of t_3. In any case, the bottleneck operator is responsible for only one class of patients, and consequently, the three matheuristics perform the same. For this reason, computational results concern only one of the threes, SPT without loss of generality.

Figs. 12 and 13 show, for each instance (x-axis) and for the two offline approaches, namely OPT and SPT, respectively the percentage difference between the load of the bottleneck operator and the makespan (%BO) and the total patient waiting time inside the exam room (WTI). As done in previous sections, an overview of KPIs relevant to solution quality is given in Table 7 where for each approach (rows), for what concerns patient waiting time, average value per patient as well as average and maximum total values are given. For what concerns %BO, the average and the maximum values over the solved instances are given.

As for the other scenarios, we observe that the matheuristic performs better than the OPT approach in terms of solution quality. Unlike in previous scenarios, however, in these instances, the gap between makespan and bottleneck operator load can be significant.

Table 7
Scenario S8: an overview of solution quality. In regard to waiting time, expressed in minutes (m), the average waiting time per patient (AvgInd) and the average total waiting time (AvgTot) are reported together with the maximum total waiting time (MaxTot) computed across scenarios. In regard to the bottleneck operator, expressed in percentage (%), average and maximum values (across scenarios) of %BO are given. Average values are computed across the scenarios.

Model	Waiting time			%BO	
	AvgInd (m)	AvgTot (m)	MaxTot (m)	Avg (%)	Max (%)
OPT	2.32	92.59	246.00	16.50	45.00
SPT	0.74	31.82	128.00	7.12	35.67

An overview of the results in terms of efficiency is given in Table 8 where for each of the two offline approaches (rows) we report the number of times each approach outperforms the other (columns Ranking), the percentage of instances for which a feasible solution has been found (%sol), the percentage of instances for which an optimal solution has been found (%opt), the average computational time, the average gap and the maximum gap across the solved instances, as well as the gap distribution. The number of times an approach ranks first includes the cases where both the approaches get the same result, which amounts to 4 in this experiment. We observe that even if SPT performs much better than OPT in terms of number of instances solved to optimality and number of instances for which a feasible solution is provided, there are 15 instances that it is not able to solve and for a significant number of instances (about 30%) the optimality gap is greater than 10%. The set of instances used in these additional experiments, even if made by only two classes of disjoint tests seems thus to be challenging to solve.

We conclude the section by reporting some computational results on the comparison between two alternative objective functions used in the offline approaches on the 169 instances of this scenario. Specifically, we compare OPT and SPT when run with a hierarchical objective function that first minimizes makespan and second minimizes patient waiting time in one case and operator idle time in the other. When operator idle time drives the second term of the objective function, OPT is not able to get a feasible solution for any of the instances, while SPT solves 71 instances with an average computational time equal to 3538 s and an average optimality gap equal to 0.8. Instead, when the second term of the objective function is guided by patient waiting time, the same figures for SPT are: for 154 instances a feasible solution is found, with an average computational time equal to 3193 s and an average optimality gap equal to 7.1. In terms of efficiency, controlling operator idle time seems to make the problem more difficult to solve with respect to controlling patient waiting time. There are 49

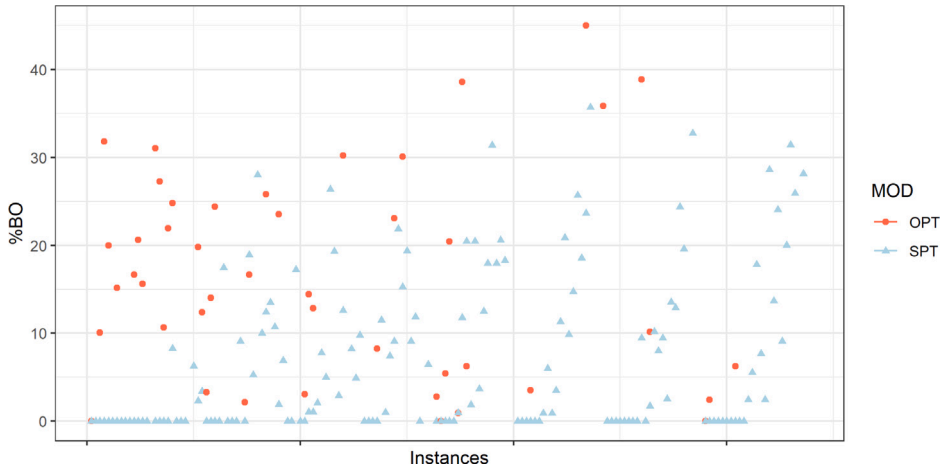

Fig. 12. Scenario S8, $H = 240$, $d_1 = 10, d_2 = 9, d_3 = 6$: percentage gap between the workload of the bottleneck operator and makespan.

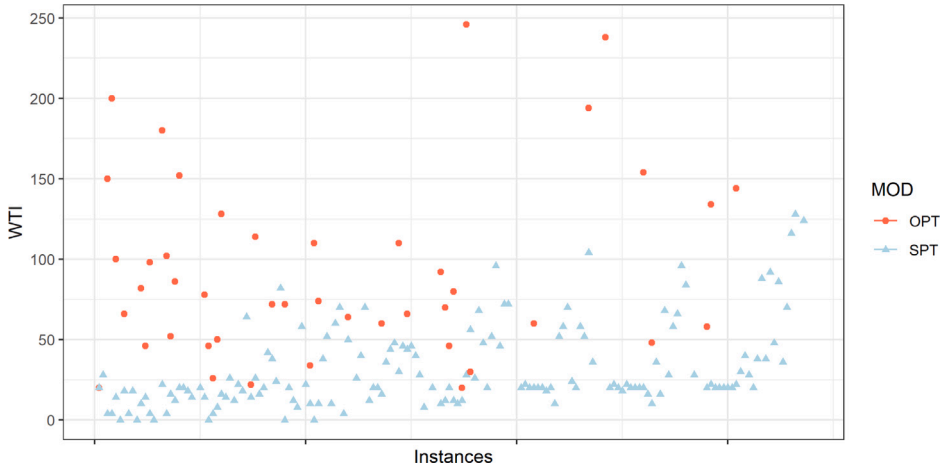

Fig. 13. Scenario S8, $H = 240$, $d_1 = 10, d_2 = 9, d_3 = 6$: Total waiting time inside exam rooms (in minutes).

Table 8
For each model, columns Ranking report the approach ranking in terms of objective function value and the number of instances for which the time limit is reached without providing any feasible solution (NotSolved). Then, efficiency information is reported: the percentage of instances for which a feasible (%sol) or optimal (%opt) solution is found, the average computational time (Mean Time), the average gap (Mean %Gap) and the maximum gap (Max %Gap) across the solved instances. Finally, information on gap distribution is given, i.e., the percentage of instances for which the optimality gap is within a given range.

Sc	NInst	Model	Ranking			%sol	%opt	Mean	Mean	Max	GapDistribution			
			1st	2nd	NotSolved			time	%Gap	%Gap	<1%	<5%	<10%	≥10%
S8	169	OPT	7	37	125	26	1	3600	16.5	45.0	9.1	22.7	31.8	68.2
		SPT	152	2	15	91	38	3193	7.1	35.6	46.1	57.1	70.8	29.2

instances in which the two objective functions provide the same value for the makespan. For each of these instances (x-axis), Fig. 14 reports on the y-axis the variation of patient waiting time (blue dots) and operator idle time (green dots) when the different objective functions are used. Interestingly, we can observe that the deterioration of patient waiting time when the second component of the objective function is guided by operator idle time is much smaller than the deterioration of operator idle time when the second component of the objective function is guided by patient waiting time. More detailed information on such deterioration is given in Table 9 where the mean value, the standard deviation, minimum and maximum values of the variation are given both for patient waiting time (WTI) in the first four columns and for operator idle time (OIT) in the last four columns. These results seem thus to reveal that considering the two criteria jointly could identify an interesting line of future research.

Table 9
Scenario S8: descriptive statistics (mean, standard deviation, minimum and maximum) of the variation of patient waiting time inside exam room (WTI) and operator idle time (OIT) in minutes, when different criteria drive the second term of the objective function — patient waiting time vs. operator idle time.

Variation of WTI				Variation of OIT			
mean	sd	min	max	mean	sd	min	max
37.71	13.12	12.00	78.00	−117.84	53.06	−268.00	−8.00

5.6. Summary results

Table 10 summarizes, in the first column, the main characteristics of the scenarios (scenario, duration and presence of the longest test t_1, planning horizon length), in the second column, the number of instances in each scenario (NInst), for columns OPT, SPT, LPT and

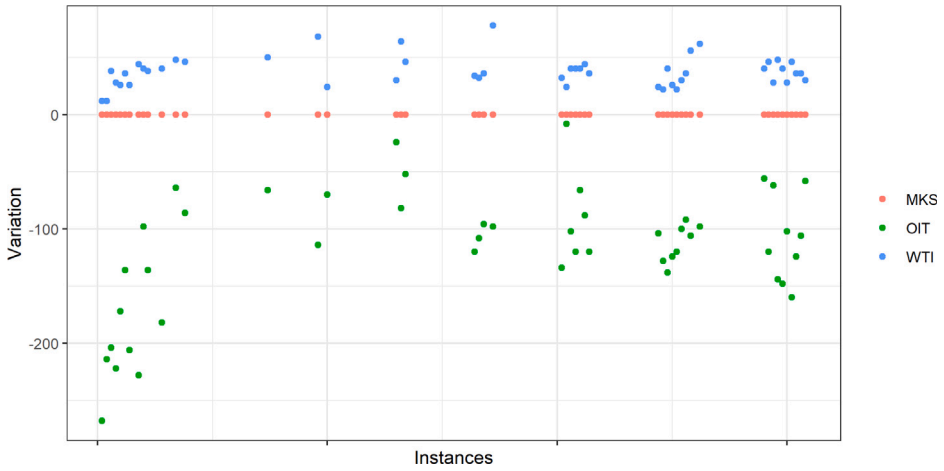

Fig. 14. Scenario S8, $H = 240$, $d_1 = 10, d_2 = 9, d_3 = 6$: for instances characterized by the same makespan, the variation of patient waiting time inside exam rooms (WTI) and operator idle time (OIT) (in minutes) when different criteria drive the second term of the objective function — patient waiting time vs. operator idle time. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 10
Summary of experiments and results: number of instances for each scenario (NInst) and percentage of instances for which each model (OPT, LPT, SPT, RR) found a feasible solution, percentage of simulation runs for which the online approaches (FSA, FRA) led to overtime.

Scenario	NInst	OPT	SPT	LPT	RR	FSA	FRA
S1 - $d_1 = 28, H = 480$	98	0.0	100	100	100	70.0	99.7
S2 - $d_1 = 28, H = 240$	69	100.0	100	100	100	2.2	71.7
S3 - $d_1 = 14, H = 240$	1742	43.2	100	100	100	23.2	90.1
S4 - $d_1 = 14, t_1 \notin C_1, H = 240$	100	79.0	99	100	100	0.0	41.0
S5 - $d_1 = 14, t_1 \notin C_2, H = 240$	100	49.0	100	100	100	0.0	37.0
S6 - $d_1 = 14, t_1 \notin C_3, H = 240$	100	55.0	99	100	99	0.0	37.0
S7 - $d_1 = 14, t_1 \notin C_4, H = 240$	100	61.0	100	100	100	0.0	38.0

RR, the percentage of instances for which each model found a feasible solution, while, for the online approaches FSA and FRA, the table shows the percentage of simulation runs for which the last patient left the PAT clinic after the scheduled closing time (overtime).

Fig. 15 summarizes the results obtained on the seven scenarios S1–S7 (x-axis) in terms of optimality gap, average percentage difference between makespan and workload of the bottleneck operator, average makespan, as well as average total waiting time inside the exam room.

We conclude the section by summarizing the main findings supported by the experiments. For scenarios S1–S3, the three matheuristics are always able to determine the optimal solution in terms of makespan. For scenarios S4–S7, the three matheuristics determine the optimal makespan in the majority of the cases and there is at least one offline approach that provides the optimal makespan. Offline approaches provide very small waiting times. Among the four offline policies, RR ranks first or second in 91% of the cases, while SPT and LPT rank first or second in 43% of the cases for scenarios S4–S7. Hence, we conclude that RR performs best, while SPT and LPT seem to be quite equivalent. All three matheuristics are computationally efficient. A possible explanation for the soundness of the policy RR is that it is the one policy that, at any given time, maintains the greatest diversity of the mix of patients currently served. In this way, all operators tend to be active at the same time and can work in parallel. In contrast, each of the other policies tends to serve patients of the same class at the same time. If patients of the class with the smallest test subset were numerous, in a given time interval only a sub-set of operators could be active, creating the conditions for many temporary deadlock situations to occur. We also observe that the presence of the longest test in each patient class and the minimization of patient waiting time as a secondary objective seem to have a positive impact on the ability of offline approaches to find feasible solutions. For the online policies, makespan always exceeds the opening time and the overtime can be high for FRA. FSA performs significantly better than FRA in terms of makespan. On the contrary, FRA is significantly better than FSA in terms of waiting time inside the exam room.

Now, we have all the elements to argue that the particular assumption we made on the number of rooms, i.e., $n_R = n_O$, is well substantiated and based on sound motivations. About the other settings, the following are some considerations. Regarding the $n_R > n_O$ option, on the one hand, at each instant either some rooms are empty or at least $n_R - n_O$ patients are occupying a room while receiving no service, potentially shifting patient waiting time from outside the room to inside the room [39]. On the other hand, this option makes temporary deadlocks less likely to occur since the room constraint becomes less binding. Therefore, human resources could be better exploited [39] and makespan could improve. On the contrary, in the $n_R < n_O$ setting, human resources (operators) would be underutilized whatever the schedule, since at least $n_O - n_R$ operators would be idle anytime. As a counter effect, WTI could decrease and service quality could increase, to the potential detriment of throughput since now, at most $n_R < n_O$ patients can be attended in parallel. As computational results showed that the proposed scheduling policies can keep WTI at bay in the $n_O = n_R$ case, we suggest that the $n_R = n_O$ option should be preferable with respect to the $n_R < n_O$ one. Although it is up to the PAT clinic manager to establish the right balance in the provision of resources, we believe that setting $n_R = n_O$ captures a very significant case worth studying it.

6. Managerial insights

A key feature of this study is to consider the *room constraint* in an outpatient PAT clinic, that is, a service model in which a patient occupies the room until all the required tests are completed and tests are administered by different operators inside the rooms sequentially, in any order. Note that patients stay inside an exam room while operators move between rooms. The service model based on the room constraint has recently gained momentum as an alternative to the *traditional* service organization where each ambulatory room is assigned to an operator and patients are the ones who move between rooms.

It is well documented that having multidisciplinary operators attending the patient in the same PAT clinic improves provider satisfaction [60]. This service model is spreading to other medical specialties as well. However, very few centers implement the room constraint feature, whose advantages and disadvantages are discussed hereafter, from the perspective of both patient and operator.

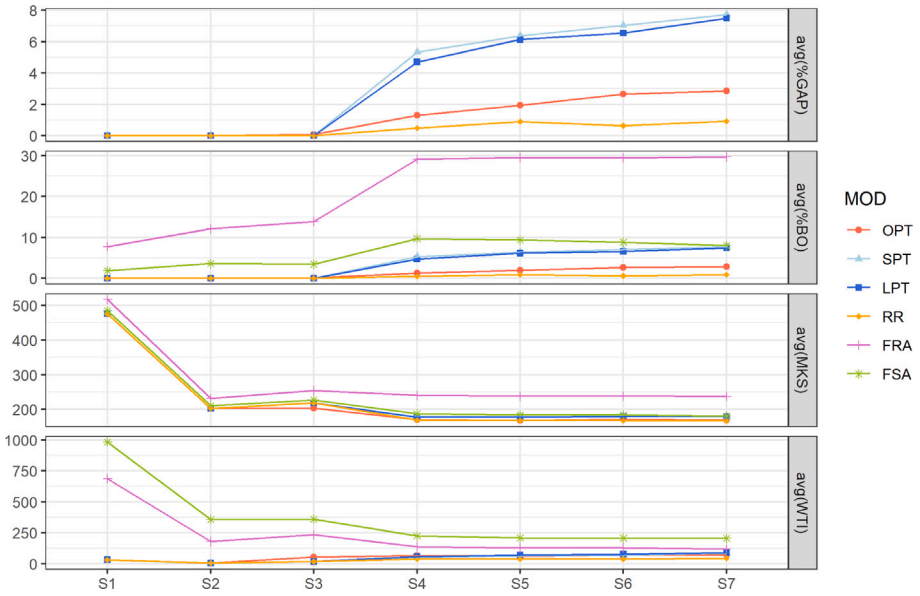

Fig. 15. An overview of results: S1–S7 scenarios.

- *Patient's perspective.* There is no doubt that concentrating all of a patient's tests in a single room has obvious advantages, especially for frail patients, as this saves the patient from having to move and to dress/undress each time they enter a new ambulatory. Nevertheless, the time spent in the common waiting room together with other patients requiring similar services may foster socialization [61] and improve patient's satisfaction compared to waiting alone in the exam room for the next operator to arrive. Overall, on the patient side, designing a centralized and leaner service that concentrates all tests in one room seems to offer more advantages than disadvantages. A recent study [62] reports interesting results from patient shadowing at a primary care clinic to identify both best practices and opportunities for improvement, and to foster patient engagement and activation. Among the key findings, two are also valuable in the PAT clinic setting. First, the study reports that even for returning patients it is highly beneficial to know in advance the sequence of tests they have to undergo and how long the entire process will take, since this allows to properly orient the patient to the process, relieving anxiety or confusion. This comes in favor of offline scheduling policies. Second, the patient is more likely to get disoriented when having to move between different exam rooms, which supports the one-room practice characterizing a PAT clinic. Finally, thanks to the room constraint, patients avoid multiple queues, one at each ambulatory room. In brief, we may say that the the room constraint improves patient's satisfaction.
- *Operator's perspective.* Compared to a traditional service organization, the room constraint may cause operator some discomfort related to repeatedly moving across rooms, which may be slightly mitigated by scheduling as the first test the one the previous patient in the same room took as last. Note that the room constraint (i) provides opportunities for information-sharing among operators on a same patient, which is particularly crucial when treating patients with specific needs as paediatric patients [63] and (ii) patients need not be shepherded to the another room after each test since they stay in the same room; (iii) finally, room constraint limits the gridlock in hallways and operators' frustration of finding patients in areas where they should not be.

The presence of room constraint has important implications on system performance. Specifically, it impacts on system throughput (number of patients served per unit time). When the number of rooms equals the number of operators, the number of busy operators does not necessarily coincide with the number of occupied rooms: the latter is greater than the former anytime a roomed patient is waiting for an operator who is busy in another room (the room deadlock introduced in Section 2). Consequently, an operator may be idle even if there are unattended patients in need of their services but no free rooms. Thus, rooms become bottleneck resources whose utilization needs to be optimized to increase system throughput and reduce patient waiting time.

The room constraint may also cause a fragmentation of operator idle time. In our study, operators are available for the entire period covered by the makespan, therefore makespan minimization also reduces their idle time. Operator idle time and patient waiting time are potentially conflicting: without the room constraint, the former could be kept to zero to the detriment of the latter, while with the room constraint, when minimizing the makespan, a trade-off between the two can be reached.

The room constraint in a PAT clinic makes patient scheduling more difficult. Scheduling problems with the room constraint were quantitatively evaluated only either in presence of simple online booking policies [37] or when patient workflow is almost fixed [36] and thus test sequencing is not an issue. In a traditional organization a rotation schedule would return an optimal schedule whose makespan equals the load of the bottleneck operator, but patients may experience long waiting times between tests (as in Fig. 4).

The room constraint is also challenging from a modeling perspective. We proposed a MILP model whose solution is computationally demanding when instances reach real-life sizes. We also proposed three effective and efficient heuristics based on bottleneck operator scheduling. Among them, the *Round Robin* heuristic outperformed the other two. Therefore, a room constraint based PAT clinic can be efficiently managed provided that solution tools as the ones discussed here are adopted. They help deliver patient-centered service (with all the above listed patient advantages) whose makespan often equals the one yielded by the Rotation Scheduling algorithm, along with low patient waiting time, providing an excellent tradeoff between throughput and service quality.

Finally, as the quantitative impact of the room constraint cannot be evaluated a priori for a given instance but only a posteriori, the exhaustive generation of instances for a variety of realistic settings provided in this paper allows the service provider to evaluate the KPIs

obtainable depending on the realization of the patient mix (what-if analysis).

We conclude by saying that, from a managerial point of view, the room constraint seems to offer numerous advantages for both patients and operators, provided that ad hoc solution approaches are used to solve the resulting challenging scheduling problem. This is the price to be paid if aiming at high service quality and high system throughput.

7. Conclusions

This paper identifies a new machine scheduling problem as the core of the offline appointment scheduling process for PAT with the room constraint, i.e., a Proportionate Open Shop Problem with renewable resources and missing operations. A network-flow-based MILP model has been proposed, which paves the way for further generalizations, such as multi-skilled operators. The patient-wise and the system-wise performance of the offline approach has been compared with two classical online procedures. However, none were able to provide a satisfactory compromise between computational complexity and solution quality. A core set of decisions has been identified in the mathematical model, which characterize optimal or suboptimal solutions, and which, once taken, define a much simpler MILP subproblem that experimentally proved to be quickly solvable for realistic instances. Such decisions involve the sequence of activities of the bottleneck operator. Three matheuristics have been proposed which take such decisions guided by as many different greedy criteria, while the remaining decisions are left to the solver. Since one feature of our problem is that some jobs involve just subsets of operations, these matheuristics have been tested on different scenarios where the bottleneck operator is not present in all the jobs. All matheuristics proved robust with respect to these variants. One stands out as the best-performing one, thus proving its viability for tackling realistic instances.

This work is not without limitations. While we find that the literature supports the hypothesis of considering test durations independent of patients, when analyzing the distributions of service times that characterize the case study that inspired us, we note non-negligible standard deviation values. A finer patient stratification that makes the duration of service dependent on certain patient characteristics (e.g. age, comorbidity, frailty) could be a useful tool to make the classes of patients more homogeneous. Another cause of inhomogeneity of service times could be the presence of operator-dependent (i.e., more or less experienced) test delivery times. Operator-dependent duration could be managed in our approach by simply varying the test duration parameters according to which operator will deliver the test. The MILP model proposed here supports this generalization.

We believe that the efficient solution approaches presented here, which have been developed to target our case, may well provide the building blocks of more elaborate approaches needed to tackle more general settings, such as $n_O > n_T$ and multiple, cross-trained operators, as well as a robust version of the problem.

CRediT authorship contribution statement

Saligrama Agnihothri: Conceptualization, Formal analysis, Writing – original draft, Writing – review & editing. **Paola Cappanera:** Conceptualization, Methodology, Software, Validation, Formal analysis, Writing – original draft, Writing – review & editing. **Maddalena Nonato:** Conceptualization, Methodology, Validation, Formal analysis, Writing – original draft, Writing – review & editing, Visualization. **Filippo Visintin:** Conceptualization, Software, Validation, Formal analysis, Writing – original draft, Writing – review & editing, Visualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Table A.11
Empirical distribution of service times. Mean service time μ and standard deviation σ are expressed in minutes. NP stands for Nurse Practitioner.

Test name	μ	σ	Best fitting service time distribution with scale and shape parameters
Nurse	27.79	10.26	Gamma (6.73, 4.14)
NP	8.83	5.10	Lognormal (2.02, 0.56)
Lab	5.98	3.11	Lognormal (1.82, 0.45)
X-ray	7.00	2.51	Lognormal (1.61, 0.56)

Data availability

Data will be made available on request.

Acknowledgment

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. The authors are grateful to the editors and reviewers for their helpful and stimulating comments.

Appendix A. A case study

This appendix provides additional details about the hospital pre-admission testing clinic (PATC) that inspired this study. The way test administration and patient flow are managed in a centralized PATC has already been described in the introduction (Section 1). Here, more information about the patient arrival process, test duration, and human and non-human resources are provided. All data were collected manually during an observation period of approximately three months.

Patient arrivals. The PATC accepts both scheduled and walk-in patients. It was observed that scheduled patients (about 60% of the total) may arrive either before or after the scheduled time. The remaining patients (about 40%) are not scheduled and arrive randomly. The resulting arrival process, both scheduled and unscheduled, is observed to follow a non-homogeneous Poisson process. Hourly arrival rates during peak period vary between 4 to 5.5 patients.

Patients flow. Patients visiting the clinic will see anywhere between three and six skilled operators depending on the patient's needs and their surgeon's recommendation regarding the procedure. Before being taken to an exam room, every patient sees the pharmacist first, to provide information about their allergies and medications they are taking. Tests that may need to be performed in the exam room include blood collection (Lab), X-rays, EKG, meeting with a nurse practitioner (NP) if undergoing anesthesia, and meeting with a registered nurse (Nurse) who collects medical history and informs patients what to expect on the surgery date. Fig. A.16 depicts the patient flow process at the PATC.

Human and non-human resources. Human resources include different types of nurses and technicians. Non-human resources include the exam rooms and medical equipment such as EKG and X-ray machines. The PAT clinic staff is made of five registered nurses, one nurse practitioner, two X-ray technicians and two Lab technicians. Lab and X-ray technicians are cross trained. The former draw blood, the latter operate the X-ray machine, and all of them may operate the EKG machine. There is only one X-ray room; if a patient needs an X-ray, an X-ray technician must retrieve the patient from their exam room and take them to the X-ray room. There is only one EKG machine as well. However, it may be shared by other offices in the building. Whenever a patient needs an EKG test, if the EKG machine is available, the technician doing the test retrieves it and brings it to the exam room where the test is carried out. If the EKG machine is being used, either in another exam room or out of the PATC, the technician must wait for the machine to become available and then retrieve it before delivering the test.

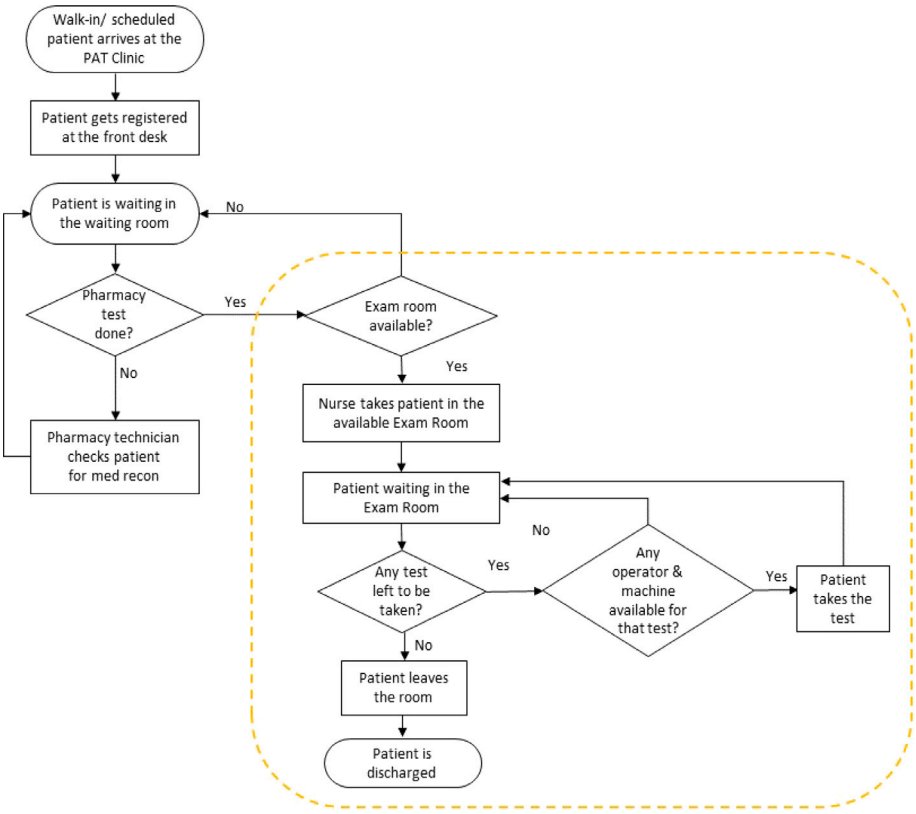

Fig. A.16. Patient flow chart at the PAT clinic. The yellow dotted line encapsulates the core of the scheduling process that inspired this paper.

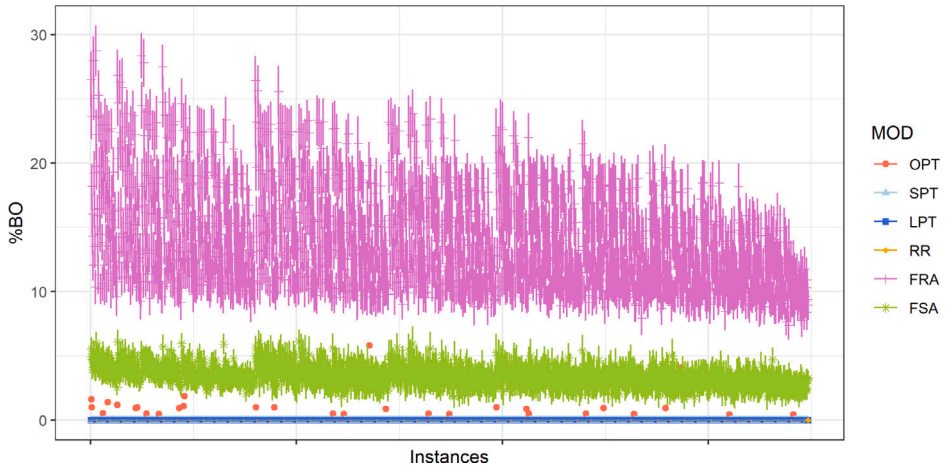

Fig. B.17. Scenario S3, $H = 240$, $d_1 = 14$: percentage gap between the workload of the bottleneck operator and makespan.

The Tests. In our study, visiting the pharmacist happens outside the exam room. Since the objective of the study is to investigate the extent the exam rooms, intended as renewable resources (see Section 3), affect the scheduling, we are focusing on the activities performed only inside exam rooms. Therefore, the interview with the pharmacist has not been considered. Moreover, the recorded processing times for EKG test include the actual time to perform the test and any waiting time needed to retrieve the machine. Therefore, the recorded time may not be a good proxy for the EKG test duration. Hence, the EKG test was not included in our analysis.

Regarding all the other tests, the data have been grouped based on the test, disregarding which operator delivered the test, in case multiple operators are entitled to perform it.

Thus, we considered four tests (Nurse test, Nurse Practitioner test, Lab test, and Xray test). For each of the four tests, the empirical distribution of the service times has been fitted. Details are provided in Table A.11. The duration of the tests provided in Table 3 (scenario S1) were obtained by rounding the mean service time μ in Table A.11 to the nearest integer.

To maintain the same structure as in the real case, where the number of tests coincides with the number of exam rooms, in this paper we assume that there are 4 exam rooms. Consequently, the daily opening time has been reduced from 10 to 8 h.

Appendix B. Further computational results

This appendix contains figures concerning solution quality for scenarios S3–S7. Specifically, considering scenario S3, Fig. B.17 shows the

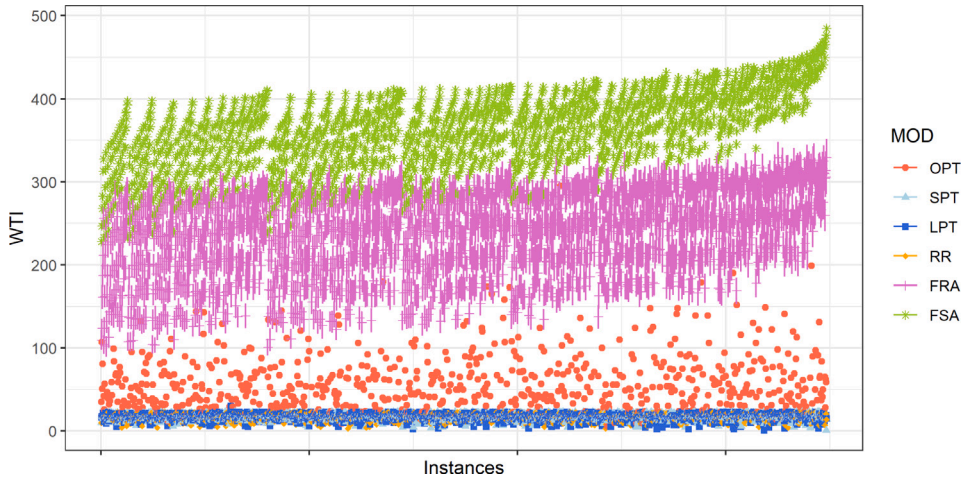

Fig. B.18. Scenario S3, $H = 240$, $d_1 = 14$: Total waiting time inside exam rooms (in minutes).

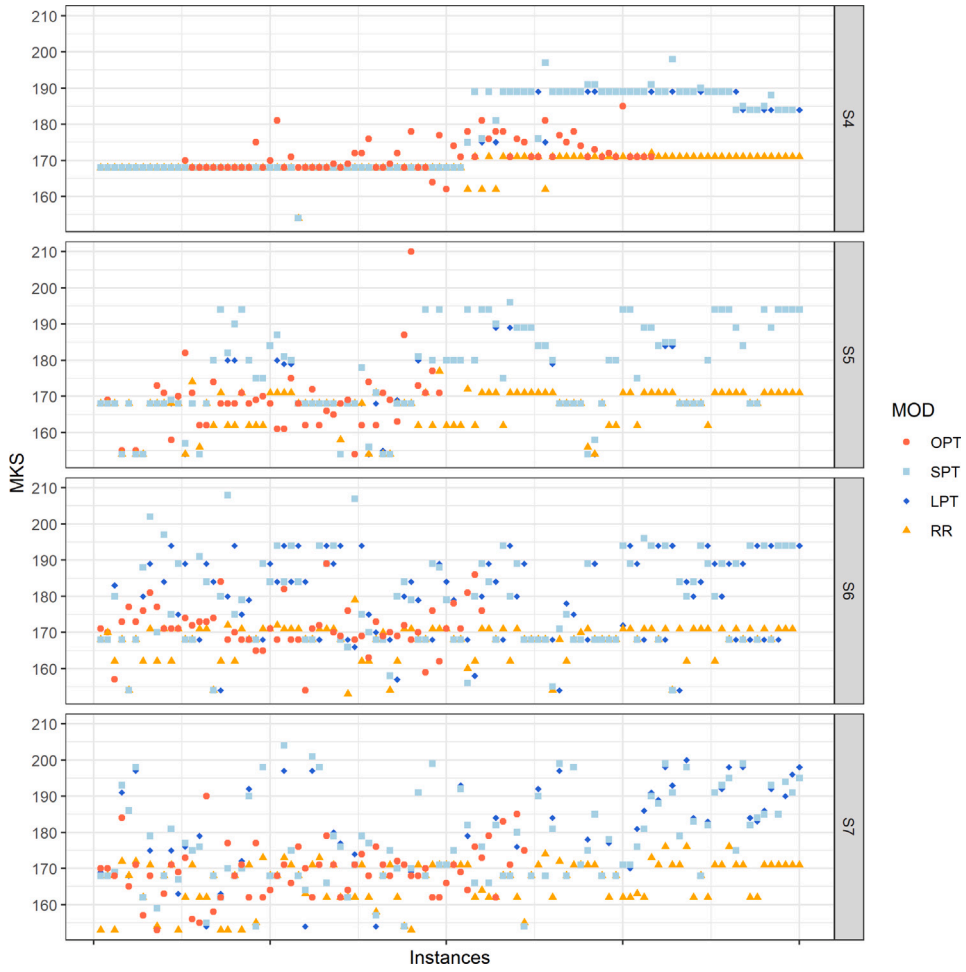

Fig. B.19. Scenarios S4–S7, $H = 240$: Makespan (in minutes).

percentage difference between the workload of the bottleneck operator and the makespan, while Fig. B.18 reports the waiting time inside the exam room (for FRA and FSA the plot shows the 95% confidence intervals for the mean obtained with 30 simulation runs).

Further results are reported for scenarios S4–S7. As expected, offline approaches outperform online ones in terms of all the KPIs used. For this reason, the following figures focus on the comparison between the offline policies only, i.e., OPT, SPT, LPT and RR. Figs. B.19, B.20, and B.21 report separately for each scenario and for each instance, respectively the makespan, the percentage difference between makespan and workload of the bottleneck operator, and waiting time inside the exam room.

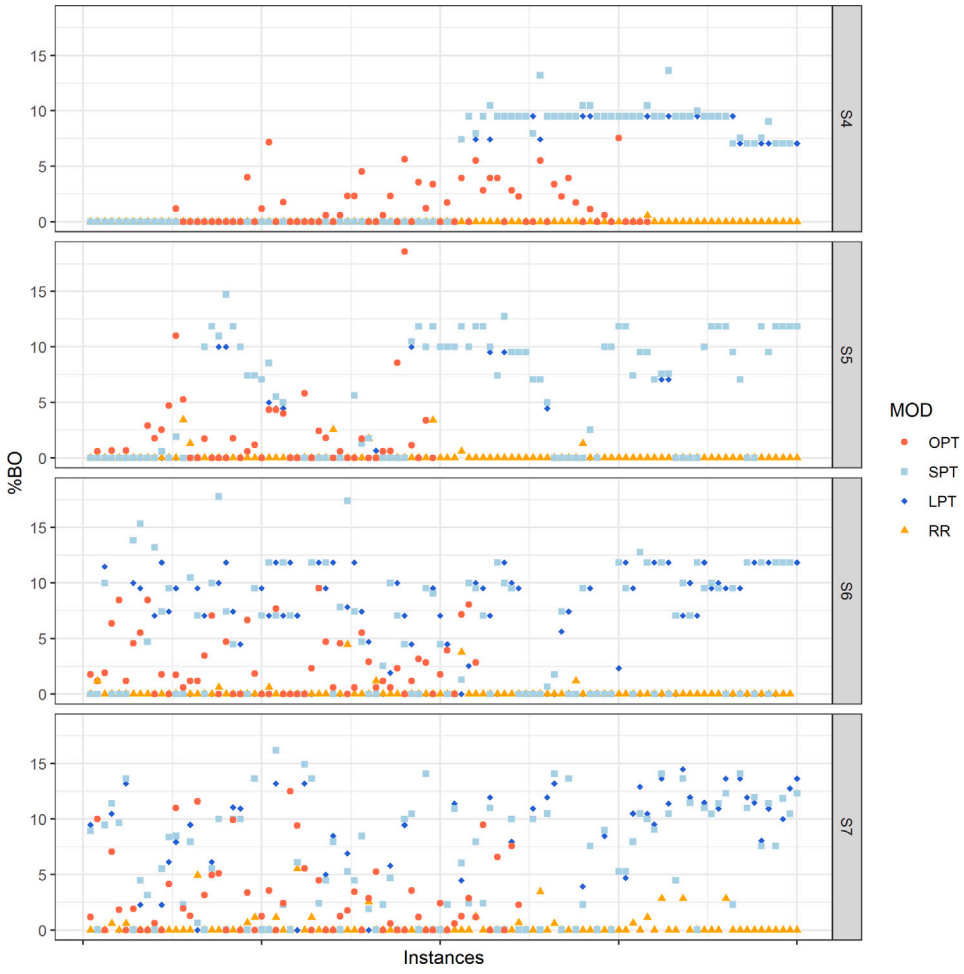

Fig. B.20. Scenarios S4–S7, $H = 240$: percentage gap between the workload of the bottleneck operator and makespan.

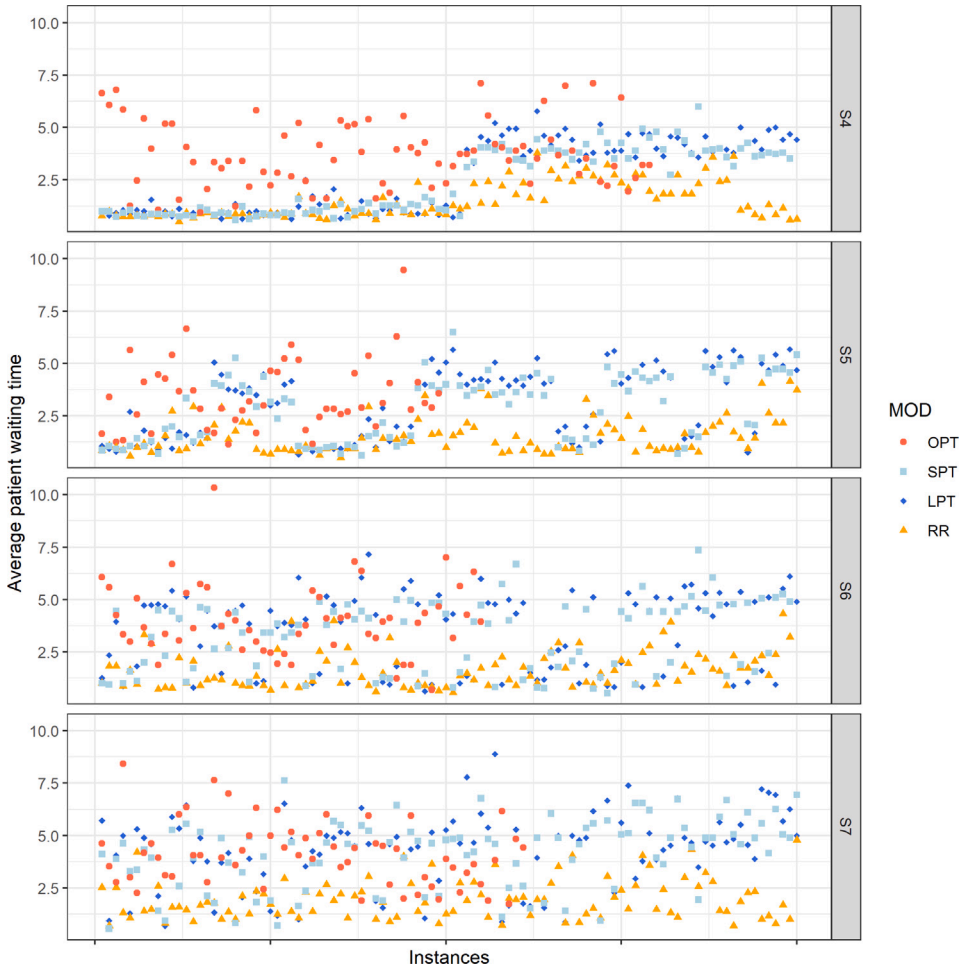

Fig. B.21. Scenarios S4–S7, $H = 240$: Average patient waiting time inside exam room (in minutes).

References

[1] Emanuel A, Macpherson R. The anaesthetic pre-admission clinic is effective in minimising surgical cancellation rates. Anaesth Intensive Care 2013;41(1):90–4.

[2] Ferschl MB, Tung A, Sweitzer B, Huo D, Glick DB. Preoperative clinic visits reduce operating room cancellations and delays. Anesthesiology 2005;103:855–9.

[3] van Klei WA, Moons KG, Rutten CL, Schuurhuis A, Knape JT, Kalkman CJ, Grobbee DE. The effect of outpatient preoperative evaluation of hospital in-patients on cancellation of surgery and length of hospital stay. Anesth Analg 2002;94:644–9.

[4] Cordier JP, Riane F. Towards a centralised appointments system to optimise the length of patient stay. Decis Support Syst 2013;55(2):629–39.

[5] Morrice DJ, (Ester) Wang D, Bard JF, Leykum LK, Noorily S, Veerapaneni P. A patient-centered surgical home to improve outpatient surgical processes of care and outcomes. IISE Trans Healthc Syst Eng 2014;4:119–34.

[6] Creemers S, Lambrecht MR, Beliën J, Van den Broeke M. Evaluation of appointment scheduling rules: A multi-performance measurement approach. Omega - Int J Manage Sci 2021;100:102231. http://dx.doi.org/10.1016/j.omega.2020.102231.

[7] Bailey NTJ. A study of queues and appointment systems in hospital out-patient departments, with special reference to waiting-times. J R Stat Soc Ser B Stat Methodol 1952;14(2):185–99.

[8] Welch JD, Bailey NTJ. Appointment systems in hospital outpatient departments. Lancet 1952;259:1105–8.

[9] Fetter RB, Thompson JD. Patients' waiting time and doctors' idle time in the outpatient setting. Health Serv Res 1966;1:66–90.

[10] Cayirli T, Veral E, Rosen H. Designing appointment scheduling systems for ambulatory care services. Health Care Manag Sci 2006;9(1):47–58.

[11] Cayirli T, Veral E, Rosen H. Assessment of patient classification in appointment system design. Prod Oper Manag 2008;17(3):338–53.

[12] Pan X, Geng N, Xie X, Wen J. Managing appointments with waiting time targets and random walk-ins. Omega - Int J Manage Sci 2020;95:102062. http://dx.doi.org/10.1016/j.omega.2019.04.005.

[13] Deceuninck M, Fiems D, De Vuyst S. Outpatient scheduling with unpunctual patients and no-shows. Eur J Oper Res 2018;265(1):195–207.

[14] Kaandorp GC, Koole G. Optimal outpatient appointment scheduling. Health Care Manag Sci 2007;10:217–29.

[15] Vanden Bosche PM, Dietz DC, Simeoni JR. Scheduling customer arrivals to a stochastic service system. Nav Res Logist 1999;46:549–59.

[16] Srinivas S, Choi S. Designing variable-sized block appointment system under time-varying no-shows. Comput Ind Eng 2022;172(Part A):108596. http://dx.doi.org/10.1016/j.cie.2022.108596.

[17] Denton B, Gupta D. A sequential bounding approach for optimal appointment scheduling. IIE Trans 2003;35:1003–16.

[18] Robinson LW, Chen RR. Scheduling doctors' appointments: optimal and empirically-based heuristic policies. IIE Trans 2003;35:295–307.

[19] Liu L, Liu X. Block appointment systems for outpatient clinics with multiple doctors. J Oper Res Soc 1998;50(9):877–91.

[20] Liao C, Pegden CD, Rosenshine M. Planning timely arrivals to a stochastic production or service system. IIE Trans 1993;25:36–73.

[21] Klassen KJ, Yoogalingam R. Improving performance in outpatient appointment services with a simulation optimization approach. Prod Oper Manag 2009;18(4):447–58.

[22] Klassen KJ, Yoogalingam R. Appointment scheduling in multi-stage outpatient clinics. Health Care Manag Sci 2019;22:229–44.

[23] Golmohammadi D, Zhao L, Dreyfus D. Using machine learning techniques to reduce uncertainty for outpatient appointment scheduling practices in outpatient clinics. Omega - Int J Manage Sci 2023;120:102907. http://dx.doi.org/10.1016/j.omega.2023.102907.

[24] Shao K, Fan W, Lan S, Kong M, Yang S. A column generation-based heuristic for brachytherapy patient scheduling with multiple treatment sessions considering radioactive source decay and time constraints. Omega - Int J Manage Sci 2023;118:102853. http://dx.doi.org/10.1016/j.omega.2023.102853.

[25] Yu S, Kulkarni VG, Deshpande V. Appointment scheduling for a health care facility with series patients. Prod Oper Manag 2020;29(2):388–409.

[26] Cappanera P, Visintin F, Banditori C, Di Feo D. Evaluating the long-term effects of appointment scheduling policies in a magnetic resonance imaging setting. Flex Serv Manuf J 2019;31:212–54.

[27] Cappanera P, Gavanelli M, Nonato M, Roma M. A decomposition approach to the clinical pathway deployment for chronic outpatients with comorbidities. In: Optimization in artificial intelligence and data sciences, proceedings of ODS-2021. AIRO springer series, 2022, p. 213–26.

[28] Cappanera P, Gavanelli M, Nonato M, Roma M. Decomposition approaches for scheduling chronic outpatients' clinical pathways in answer set programming. J Logic Comput 2023. http://dx.doi.org/10.1093/logcom/exad038.

[29] Cappanera P, Gavanelli M, Nonato M, Roma M. Logic-based benders decomposition in answer set programming for chronic outpatients scheduling. Theor Pract Log Program 2023;23(4):848–64.

[30] Cayirli T, Veral E. Outpatient scheduling in health care: A review of literature. Prod Oper Manag 2003;12(4):519–49.

[31] Ahmadi-Javid A, Jalali Z, Klassen KJ. Outpatient appointment systems in healthcare: A review of optimization studies. Eur J Oper Res 2017;258(1):3–34.

[32] Marynissen J, Demeulemeester E. Literature review on multi-appointment scheduling problems in hospitals. Eur J Oper Res 2019;272(2):407–19.

[33] Pazoki M, Samarghandi H. Regulating patient care in walk-in clinics. Omega - Int J Manage Sci 2021;99:102200.

[34] Ferreira DC, Marques RC, Nunes AM, Figueira JR. Patients' satisfaction: The medical appointments valence in portuguese public hospitals. Omega - Int J Manage Sci 2018;80:58–76.

[35] Yokouchi M, Aoki S, Sang H, Zhao R, Takakuwa S. Operations analysis and appointment scheduling for an outpatient chemotherapy department. In: Proceedings of the 2012 winter simulation conference (WSC). IEEE; 2012, http://dx.doi.org/10.1109/WSC.2012.6464990.

[36] Zhang P, Bard JF, Morrice DJ, Koenig KM. Extended open shop scheduling with resource constraints: Appointment scheduling for integrated practice units. IISE Trans 2019;51(10):1037–60.

[37] Agnihothri S, Banerjee A, Thalacker G. Analytics to improve service in a pre-admission testing clinic. In: 48th P Ann HICSS. 2015, p. 1325–31.

[38] Agnihothri S, Visintin F, Banerjee A. Simulating a hospital preadmission testing center to improve patient service. In: The best thinking in business analytics. Pearson FT Press; 2015, p. 1–12.

[39] White DL, Froehle CM, Klassen KJ. The effect of integrated scheduling and capacity policies on clinical efficiency. Prod Oper Manag 2011;20(3):442–55.

[40] Zhou S, Li D, Yin Y. Coordinated appointment scheduling with multiple providers and patient-and-physician matching cost in specialty care. Omega - Int J Manage Sci 2021;101:102285. http://dx.doi.org/10.1016/j.omega.2020.102285.

[41] Hu M, Xu X, Li X, Che T. Managing patients' no-show behaviour to improve the sustainability of hospital appointment systems: Exploring the conscious and unconscious determinants of no-show behaviour. J Clean Prod 2020;269:122318. http://dx.doi.org/10.1016/j.jclepro.2020.122318.

[42] Matta ME, Elmaghraby SE. Polynomial time algorithms for two special classes of the proportionate multiprocessor open shop. Eur J Oper Res 2010;201(3):720–8.

[43] Zhang J, Wang L, Xing L. Large-scale medical examination scheduling technology based on intelligent optimization. J Comb Optim 2019;37(1):385–404.

[44] Chen B, Potts C, Woeginger G. A review of machine scheduling: Complexity, algorithms and approximability. In: Handbook of combinatorial optimization. Boston, MA: Springer; 1998, p. 1493–641.

[45] Anand E, Panneerselvam R. Literature review of open shop scheduling problems. Intell Inf Manag 2015;7:33–52.

[46] Ahmadian MM, Khatami M, Salehipour A, Cheng TCE. Four decades of research on the open-shop scheduling problem to minimize the makespan. Eur J Oper Res 2021;295(2):399–426.

[47] Graham RL, Lawler EL, Lenstra JK, Rinnooy Kan AHG. Optimization and approximation in deterministic sequencing and scheduling: a survey. Ann Discrete Math 1979;5:287–326.

[48] Hefetz N, Adiri I. A note on the influence of missing operations on scheduling problems. Nav Res Logist Q 1982;29(3):535–9.

[49] Adiri I. Open-shop scheduling problems with dominated machines. Nav Res Logist 1989;36(3):273–81.

[50] Fiala T. An algorithm for the open-shop problem. Math Oper Res 1983;8:100–9.

[51] Sevastianov SV. Nonstrict vector summation in multi-operation scheduling. Ann Oper Res 1998;83:179–212.

[52] Kubiak W. Proportionate and ordered open shops. In: A book of open shop scheduling. Springer International Publishing; 2022, p. 165–92.

[53] Dror M. Openshop scheduling with machine dependent processing times. Discrete Appl Math 1992;39(3):197–205.

[54] Naderi B, Zandieh M, Yazdani M. Polynomial time approximation algorithms for proportionate open-shop scheduling. Int Trans Oper Res 2014;21(6):1031–44.

[55] Matta ME. A genetic algorithm for the proportionate multiprocessor open shop. Comput Oper Res 2009;36:2601–18.

[56] Blazewicz J, Lenstra JK, Kan AR. Scheduling subject to resource constraints: classification and complexity. Discrete Appl Math 1983;5(1):11–24.

[57] Agnetis A, Flamini M, Nicosia G, Pacifici A. A job-shop problem with one additional resource type. J Sched 2011;14(3):225–37.

[58] Wang M, Sethi S, Sriskandarajah C. Minimizing makespan in flowshops with pallet requirements. INFOR (Inf Syst Oper Res) 1996;35:277–85.

[59] Huang X. Patient attitude towards waiting in an outpatient clinic and its applications. Health Serv Manage Res 1994;7(1):2–8.

[60] Saver C. PAT makeover enhances patient and provider satisfaction. OR Manager; 2018, https://www.ormanager.com/pat-makeover-enhances-patient-provider-satisfaction/.

[61] Glover T, Parry D. A third place in the everyday lives of people with cancer: Functions of gilda's club of greater toronto. Health Place 2009;15(1):97–106.

[62] Gallan AS, Perlow B, Shah R, Gravdal J. The impact of patient shadowing on service design: Insights from a family medicine clinic. Patient Exp J 2021;8(1):88–98.

[63] Coyne I, Amory A, Gibson F, Kiernan G. Information-sharing between healthcare professionals, parents and children with cancer: More than a matter of information exchange. Eur J Cancer Care 2016;25(1):141–56.