

Schulz, Arne; Fliedner, Malte

**Article — Published Version**

## Minimizing the expected waiting time of emergency jobs

Journal of Scheduling

**Provided in Cooperation with:**

Springer Nature

*Suggested Citation:* Schulz, Arne; Fliedner, Malte (2022) : Minimizing the expected waiting time of emergency jobs, Journal of Scheduling, ISSN 1099-1425, Springer US, New York, NY, Vol. 26, Iss. 2, pp. 147-167,  
<https://doi.org/10.1007/s10951-022-00767-1>

This Version is available at:

<https://hdl.handle.net/10419/312843>

**Standard-Nutzungsbedingungen:**

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

**Terms of use:**

*Documents in EconStor may be saved and copied for your personal and scholarly purposes.*

*You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.*

*If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.*



<https://creativecommons.org/licenses/by/4.0/>



# Minimizing the expected waiting time of emergency jobs

Arne Schulz<sup>1</sup> · Malte Fliedner<sup>1</sup>

Accepted: 28 October 2022 / Published online: 12 December 2022  
© The Author(s) 2022

## Abstract

We consider a scheduling problem where a set of known jobs needs to be assigned to a set of given parallel resources such that the expected waiting time for a set of uncertain emergency jobs is kept as small as possible. On the basis of structural insights from queuing theory, we develop deterministic scheduling policies that reserve resource capacity in order to increase the likelihood of resource availability whenever an emergency job arrives. Applications of this particular scheduling problem are, for instance, found in the field of surgical operations scheduling in hospitals, where high-priority but uncertain emergencies compete for scarce operating room capacity with elective surgeries of lower priority. We compare our approaches with other policies from the literature in a comprehensive simulation study of a surgical operations unit.

**Keywords** Machine scheduling · Operating room scheduling · Non-elective surgery · Queuing theory

## 1 Introduction

In typical scheduling problems, a set of jobs  $I$  has to be scheduled on a set of given resources  $O$  such that some (often time-oriented) objective is optimized. In many planning environments, not all jobs or their characteristics are known at the time of a scheduling decision, so that there is considerable uncertainty about future demand for resources and thus about ideal resource utilization. At the heart of many scheduling problems therefore lies a form of risk management that needs to trade off the consequences of scheduling a known job on some resource against future demand induced by uncertain jobs. The importance of the scheduling decision is exacerbated if jobs cannot be interrupted once started and in particular whenever the uncertain jobs foreseeably have an emergency character, so that the success of the total system crucially depends on whether emergency jobs can be served in a timely fashion.

In this work, we study the decision problem of scheduling a known set of non-pre-emptive jobs on homogeneous resources in such a way that the waiting time of a set of uncertain emergency jobs which will be released to the system at

a later stage is kept low. Applications for this kind of planning problem have been extensively studied in the context of operating room scheduling (see the survey of Cardoen et al., 2010), where all surgeries that are carried out can typically be split up into a subset of known elective surgeries, which can be planned upfront on a daily or weekly basis, and a subset of non-elective surgeries, which most often constitute emergency surgeries that are uncertain with respect to the points in time of their occurrence and resource utilization. Due to the nature of emergencies, these latter non-elective surgeries often need to be serviced as fast as possible, so that a successful scheduling policy of elective surgeries should leave sufficient free capacity to service emergencies. Related problems have also been studied in the field of machine scheduling and maintenance in order to accommodate rush orders or maintenance operations (see Sect. 2).

While it is possible to estimate the effects of stochastic resource demand and incorporate the estimates in a stochastic optimization model, in scheduling practice often simpler policies dominate that reserve a certain set of resources or a share of their productive time to the uncertain emergency jobs. In operating room scheduling, for instance, a standard practice is to reserve one or more operating rooms exclusively for emergency surgeries in order to always have available capacity for the emergency surgeries (provided that these resources have not already been seized by another emergency). This also reduces disruptions of the planned schedule comprised of known jobs, since they are effectively not com-

---

✉ Malte Fliedner  
malte.fliedner@uni-hamburg.de

Arne Schulz  
arne.schulz@uni-hamburg.de

<sup>1</sup> Institute of Operations Management, Universität Hamburg,  
Moorweidenstraße 18, 20148 Hamburg, Germany

peting for the same resources. However, this will typically come at the expense of a lower resource utilization whenever no or only few emergencies need to be processed and might be too inflexible whenever several emergencies occur one after the other. In this work, we will investigate deterministic scheduling policies that try to schedule known jobs in such a way that the expected waiting time of uncertain emergency jobs is reduced without directly having to anticipate future demand in the analysis.

For this purpose, the paper is structured as follows: in Sect. 2, we provide a literature overview with a focus on operating room scheduling. In Sect. 3, the problem setting is described. We also analyse a simplified queuing model in order to derive theoretical insights into the problem structure (Sect. 3.1) and motivate a deterministic scheduling policy that arranges jobs without emergencies in a particular fashion (Sect. 3.2). We then develop two scheduling approaches in Sects. 4 (mixed-integer programme) and 5 (heuristic), which are tested in a computational study in Sect. 6. The paper closes with a conclusion in Sect. 7.

## 2 Literature review

Although the investigated problem can be found in different areas of application in principle (e.g. production and maintenance scheduling), the presented literature review focuses on operating room scheduling since it is in this field of enquiry that the trade-offs between the interests of elective and non-elective patients have been studied most intensely (cf. Cardoen et al., 2010 and Samudra et al., 2016). Furthermore, we will contrast our policy with two approaches that have been developed explicitly in the context of operating room scheduling (Wullink et al., 2007 and van Essen et al., 2012). Nevertheless, we give some brief references to other areas of application at the end of this section.

Operating room scheduling needs to consider a fundamental conflict between elective surgeries (jobs of normal priority), whose characteristics are known at the time of the scheduling decision, and non-elective surgeries, which typically arrive as emergency surgeries and thus have a very high priority, but are otherwise uncertain with respect to their characteristics and times of arrival (see van Riet and Demeulemeester 2015 for an in-depth discussion of the conflict of interest between elective and non-elective patients). Although non-elective surgeries are divided into urgent surgeries, which need care within a fixed time horizon of a couple of hours, and emergencies, which need care immediately, we focus only on emergencies in this paper. Both types of surgeries, electives and emergencies, have to be scheduled in a given number of operating rooms (OR) with a fixed time capacity. In order to accommodate emergency surgeries, two scheduling strategies are possible. In the first, a single or

more ORs are exclusively reserved for emergencies and elective surgeries are scheduled only in the remaining ORs. In the second, both patient groups use all ORs. In the literature, different results are found for the question whether electives and emergencies should use the same ORs or not. While Heng and Wright (2013) found out in a simulation study that ORs exclusively used by emergencies reduce the waiting time, Ferrand et al. (2014) demonstrated that emergency waiting times can be reduced significantly when both patient groups can use all ORs. To allow the shared use of the same resources by elective and emergency surgeries, a reaction plan needs to be devised that determines under what circumstances an emergency surgery that enters the system can make use of an OR that might have been assigned to an elective surgery by the scheduling procedure. Since it is typically not possible to interrupt an ongoing surgery, reaction plans often define the so-called break-in-moments (BIM), which arise whenever an elective surgery ends, and allow any waiting emergency surgery to utilize the respective operating room even if that means that an elective surgery needs to be postponed or cancelled. Wullink et al. (2007) showed that emergency waiting time will be reduced considerably whenever emergencies are operated in the OR as soon as the next surgery finishes, while maintaining a buffer at the end of the surgery day for any postponed surgery. Another approach by van der Lans et al. (2006) contains a model which distributes BIMs evenly over the surgery day. A similar approach was chosen by van Essen et al. (2012), albeit without considering buffer time.

We focus on scheduling policies minimizing the sum of emergency waiting times in this paper. Nonetheless, hospitals have to include further criteria like costs, idle time, overtime, utilization, cancellations, and lengths of stay of the patients after surgery inside the hospital in their scheduling process (cf. Beliën et al., 2009, van Veen-Berkx et al., 2016, Jung et al., 2019, Vandenberghe et al. 2020). In contrast to the sum of waiting times, Vandenberghe et al. (2019) minimized the expected maximum waiting time of emergent patients. Vandenberghe et al. (2020) minimized amongst others the risk for emergencies to suffer an excessively long waiting time. Xiao and Yoogalingam (2021) introduced a set of standby patients, who are assigned to unused capacity if a lower than expected number of emergencies arrived. Bovim et al. (2020) scheduled flexible operating room slots for emergency patients in a master surgery schedule.

Moreover, we omit further resources used by the emergency patients. In reality, further resources like surgeons, nurses, anaesthesiologists, instruments, imaging equipment (Latorre-Núñez et al., 2016) and prior and later stages, as the preoperative holding unit and the post-anaesthesia care unit, are required (Wang et al., 2015). Besides, we consider deterministic durations of elective surgeries and stochastic durations of emergencies. Amongst others, Kroer et al. (2018) consider stochastic elective durations.

**Fig. 1** The three considered scheduling policies

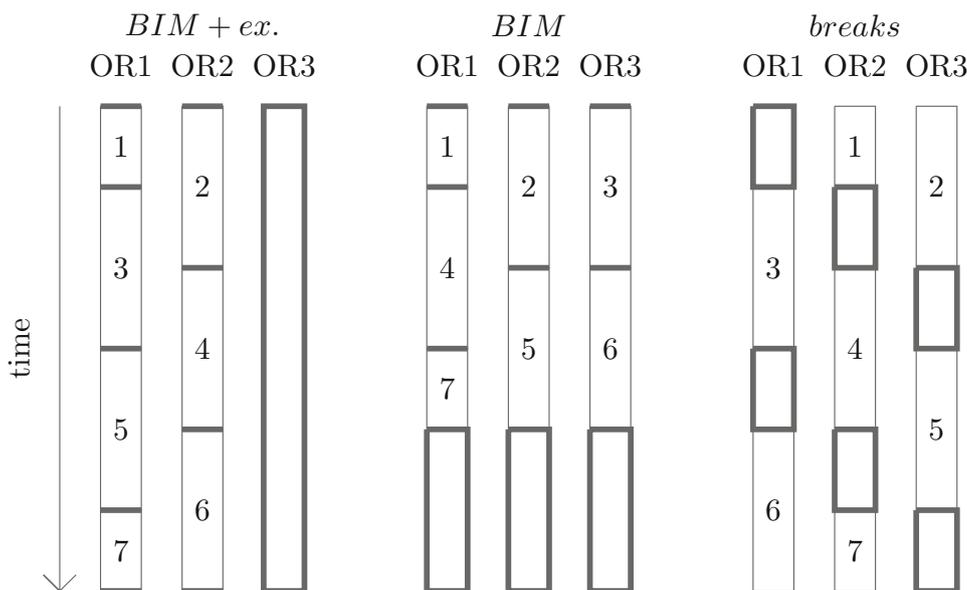


Figure 1 presents the three scheduling policies considered in this paper and their relation to the literature. The example shows seven elective surgeries which have to be scheduled in three ORs. The points in time where emergency patients can enter an OR are marked by thick solid lines and boxes, respectively. The left approach is an adapted version of the one proposed in van Essen et al. (2012). All surgeries are scheduled in the first two ORs such that the break-in-intervals (BII), i.e. the time between two consecutive BIMs, are minimal and no buffer is included. To make the comparison fair, we added an OR which is exclusively used by emergencies. Due to this extra OR, all three approaches have the same free OR time.

The second schedule in the middle is the one Wullink et al. (2007) aim at. They also schedule all elective surgeries without any buffer between but in all ORs such that the free OR time is distributed over all ORs at the end of the planning horizon. Emergency patients can enter the schedule at the end of any elective surgery. We distributed the BIIs as evenly as possible in the second approach (cf. Theorem 2) although this is not stated explicitly in Wullink et al. (2007). Furthermore, we do not consider staff requirements but assume that a surgery team is available when the surgery start is scheduled and assume that each surgery, elective and emergency, can be performed in each OR. van Essen et al. (2012) and Wullink et al. (2007) consider fixed assignments of elective patients to ORs or specialities with several assigned ORs, respectively. As further restrictions would deteriorate the approximation of the three schedules in Fig. 1, we omit them and allow each surgery to be scheduled in each OR.

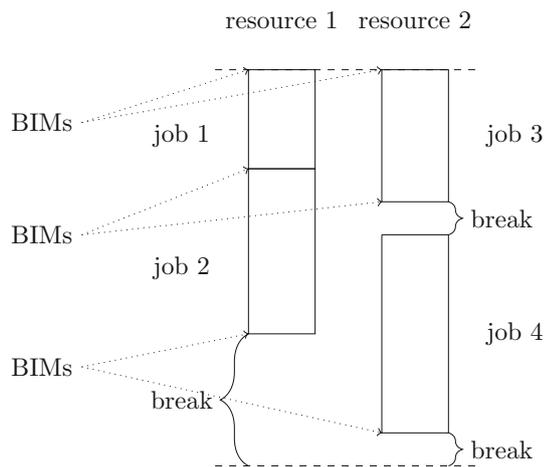
The right part of Fig. 1 shows the proposed scheduling policy of this paper. We will not only introduce BIMs at the end of any scheduled job, which can be used by emergencies to

access resource capacity, but further consider the scheduling of breaks, i.e. unreserved capacity between any two jobs on the same resource, as a possible time interval for emergencies that seek to enter the schedule.

We deviated the first two scheduling policies in Fig. 1 from the literature and established the third policy in this section. The three goals of the remaining parts of the paper are to show that these policies distribute BIMs and, in case of the third policy, also breaks appropriately over the planning horizon such that the waiting time of emergencies is as low as possible (1). Moreover, we develop scheduling approaches for the newly introduced policy (2) and evaluate all three policies in a comprehensive computational study (3). The developed approach is sufficiently general such that it can also be employed in other areas of interest. For instance, there are a variety of approaches that seek to execute rush orders that arrive at a random point in time as fast as possible (see, for instance, the rescheduling approach of Vin and Ierapetritou (2000) for batch production). A further application with similar characteristics lies in the field of planned maintenance where machine downtimes have to be scheduled in such a way as not to disrupt the production rate of the machine resources (e.g. see the review of Wang 2002). Wang et al. (2020) optimized makespan and total completion time on identical parallel machines while assuring a maximum waiting time for an emergency job.

### 3 Analysis of problem setting

We schedule a set  $I$  of non-pre-emptive jobs with known processing times  $p_i \geq 0$  for every job  $i$  on a set  $O$  of homogeneous resources over a finite planning horizon with length



**Fig. 2** Example of a schedule with break-in-moments and breaks

*T.* We are interested in finding a deterministic scheduling policy that assigns appointments to all known non-pre-emptive jobs, while indirectly considering the risk of overlong waiting times for uncertain emergency jobs that enter the system at random points in time and require immediate attention. One important application of this problem setting is found in operating room scheduling, where a number of different scheduling approaches have been developed and studied (see Cardoen et al., 2010 for a general survey of the topic).

Figure 2 presents an example for a schedule with two resources and four jobs. We define any planned downtime on a resource as a break in the schedule, which includes any time interval between the end of a job and the beginning of the next job on the same resource as well as the time between the start (end) of the planning horizon and the start (end) of the first (last) job on each resource. Any end of a job—irrespective of whether there is a break with a positive length or length zero (e.g. when a job starts immediately after the prior job on the same resource ended)—is referred to as a break-in-moment (BIM). Emergency jobs are allowed to enter the schedule and seize a resource only during breaks.

The remainder of this section is organized as follows. In Sect. 3.1, we use queueing theory to determine quality characteristics of good schedules regarding the expected waiting time of emergency jobs. Afterwards, we deduce our objectives from the theoretical results (Sect. 3.2).

### 3.1 A queueing model for distributing breaks in emergency scheduling

In order to derive basic theoretical insights into the problem structure, we will first analyse a queueing model that serves as an ideal proxy for more practice-oriented planning environments. We will then extend the results of the queueing model to identify characteristics of promising scheduling policies.

The main goal of this model is to find theoretical support for two basic intuitions with respect to the availability of resources, namely that: (1) the number of reserved resources for emergency jobs should be the same at each point in time and (2) BIMs should be distributed as evenly as possible over the time horizon.

For the purpose of our analysis in this subsection, we assume that

- the time between two emergency job arrivals is exponentially distributed with parameter  $\lambda$ , i.e. the expected time between two arrivals is  $\frac{1}{\lambda}$ .
- the processing time of emergency jobs is exponentially distributed with parameter  $\mu$ , i.e. the expected processing time is  $\frac{1}{\mu}$ .
- arrivals and processing times are independent from each other and amongst themselves.
- emergency jobs can enter the resources whenever there are breaks according to a FCFS policy.

With these assumptions, we can formulate a straightforward multi-server queueing system that will be the starting point of the analysis.

**Theorem 1** *Let  $\mu > \lambda > 0$  and  $o \in \mathbb{R}_{\geq 0}$  be the mean number of resources which are reserved for emergency jobs over the time, i.e.  $o = \frac{|O| \cdot T - \sum_{i \in I} p_i}{T}$ . Then, the expected waiting time of emergency jobs is minimal if exactly  $o$  resources are reserved for them at any point in time if  $o \in \mathbb{N}_0$  and  $\lfloor o \rfloor$  or  $\lceil o \rceil$  resources such that the mean number is  $o$  else.*

The proof of Theorem 1 is given in Appendix A.

Our next theorem states that BIMs should be distributed as evenly as possible over the time horizon. The theorem confirms the findings by van Essen et al. (2012). As we want to investigate how a schedule should be designed beforehand, we assume in the theorem that emergency jobs lead to no disruptions. Thus, we assume emergency jobs to be finished until the next BIM on the same resource. Further, no new BIM is created if an emergency job ends before the next scheduled BIM on the resource, i.e. emergencies are allowed to enter an OR only at scheduled BIMs.

**Theorem 2** *Let  $\lambda > D > 0$  be the mean time between two BIMs. Then, the expected waiting time of emergency jobs is—irrespective of the number of resources—minimal if the time between two consecutive BIMs is exactly  $D$ . If an equidistant distribution of the BIMs is not possible, waiting time of emergency jobs is minimal if the variance of the times between two successive BIMs is minimal.*

The proof of Theorem 2 is presented in Appendix B.

### 3.2 Scheduling policy

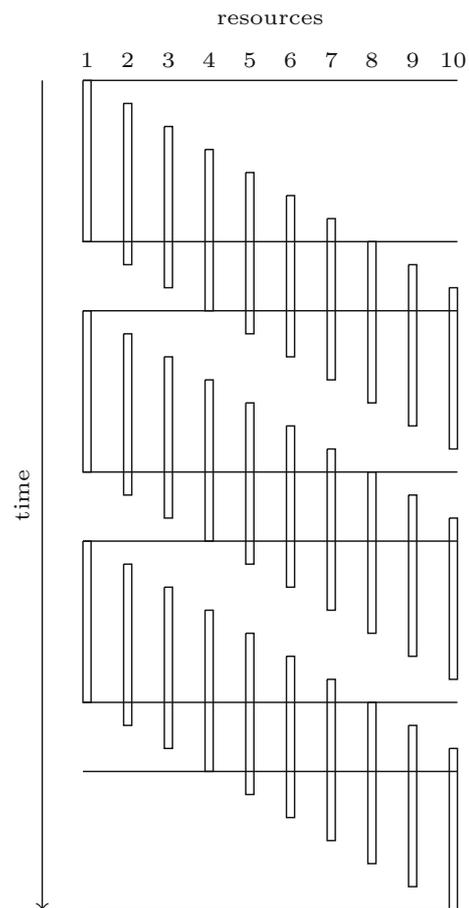
In this subsection, we derive our four objectives:

- reserve at each point in time  $\frac{|O| \cdot T - \sum_{i \in I} p_i}{T}$  resources for emergencies or deviate as little as possible from it (compare Theorem 1).
- distribute the extent of breaks evenly over the time horizon (compare Theorem 2).
- distribute the lengths of breaks, i.e. the time between the end of a job and the beginning of its successor on the same resource, evenly between any two consecutive jobs.
- utilize as many other resources as possible between the scheduling of any two jobs on the same resource, i.e. as many other resources as possible have breaks between two consecutive breaks on one resource. We name the sequence of resources that specifies where the next break is changing pattern. This leads also to an even distribution of breaks over all resources.

The analytical results of the queuing model are subject to several simplifying assumptions that are not typical for practical planning environments. Still it can be instructive to build up on these insights when considering more practice-relevant settings. As long as the arrival pattern of stochastic emergency jobs follows a stationary Poisson distribution, for instance, expected waiting times will tend to be low whenever the reserved capacity for emergency jobs is at each point in time as close to its mean as possible (objective (a)) and breaks are distributed evenly over the time horizon (objective (b)). This follows intuitively since uneven distributions of breaks over resources and/or time can give rise to bottlenecks that will lead to an increased waiting time for any emergency job that enters the system.

If we consider furthermore the situation where an emergency job is already worked on, breaks on that resource are occupied for further emergency jobs. Hence, it is advantageous if breaks have the same length (objective (c)), as no break should be particularly long, and as many other resources as possible have a break between two consecutive breaks on the same resource (objective (d)), i.e. breaks should ideally rotate always in the same sequence over all resources.

In a setting with a finite planning environment and a set of homogeneous jobs, the analytic results suggest the realization of a deterministic scheduling policy as illustrated in Fig. 3. In the depicted example, 30 homogeneous jobs are assigned to ten resources in such a way that—with exception of the beginning and the end of the schedule—three resources are always available for a break-in of any arriving emergency (objective (a)). Furthermore, the schedule satisfies further characteristics that seem beneficial: all breaks



**Fig. 3** Example for a schedule with homogeneous jobs and a finite planning horizon

between any two consecutive jobs on a resource start after the same time lag to the previous break (objective (b)), are identical in length (objective (c)), and whenever a job finishes, the same resource is only considered for another job once jobs have been assigned to all other available resources (objective (d)). Overall, this makes it less likely that an emergency job that enters a resource completely disrupts the prior schedule and reduces the risk that two consecutive emergencies will block each other on the same resource. By consecutively changing resources between breaks and extending the lengths of breaks after any job as much as possible, it is more likely that any emergency job will only interact with the least amount of scheduled jobs possible in a setting where electives and emergencies share all resources. Notice that whenever jobs are inhomogeneous (i.e. differ with respect to their processing times), it can become considerably more challenging to approximate a schedule structure as given above. On the other hand, diverging processing times open up possibilities to improve the beginning and the end of the schedule on a finite planning horizon by appropriately filling in the gaps in the schedule. Finding appropriate schedules

for inhomogeneous jobs is the subject of the next two sections.

### 4 A mixed-integer programming formulation for the scheduling problem

The mixed-integer programming (MIP) model takes a set of elective jobs  $I$  with varying processing times  $p_i$  that are to be scheduled on a set of homogeneous resources  $O$ . It seeks to find a feasible elective schedule that assigns all jobs to a resource and to a point in time while observing the four objectives stated above.

sets

$i, j, k \in I$  jobs  
 $o \in O$  resources ( $O = |O|$ )

parameters and scalars

$p_i$  processing time of job  $i$   
 $\varepsilon$  small positive number  
 $M$  big number  
 $T$  available time per resource (length of time horizon)  
 $\alpha, \beta, \gamma, \delta \in [0, 1]$  weights for the different parts of the objective

continuous variables

$a$  ensures the right changing pattern for the last job  
 $b_i$  duration of the break-in-interval that follows  $i$   
 $\bar{b}$  max duration of break-in-intervals  
 $\underline{b}$  min duration of break-in-intervals  
 $d_o$  smallest break between two consecutive jobs on resource  $o$   
 $s_i$  start time of job  $i$   
 $ov_{ij}$  amount of time that the executions of jobs  $i$  and  $j$  overlap each other  
 $u_{io}$  number of jobs which start between the start of the previous job on resource  $o$  and the start of job  $i$

binary variables

$c_i$  is 1 if the successor of job  $i$  starts at time zero and 0 else  
 $n_o$  is 1 if at least two jobs are scheduled on resource  $o$  and 0 else  
 $v_{ijo}$  is 1 if job  $j$  starts before job  $i$  and  $j$  is assigned to resource  $o$  and 0 else  
 $w_{ij}$  is 1 if jobs  $i$  and  $j$  overlap the whole processing time of  $j$  and 0 else  
 $x_{ij}$  is 1 if job  $i$  starts before job  $j$  and 0 else

$y_{io}$  is 1 if job  $i$  is scheduled on resource  $o$  and 0 else  
 $z_{ij}$  is 1 if job  $i$  ends before job  $j$  starts and 0 else

$$\min \underbrace{\alpha \cdot \sum_{i,j \in I: i \neq j} ov_{ij}}_{\text{overlap (see (a))}} + \underbrace{\beta \cdot (\bar{b} - \underline{b})}_{\text{evenly distributed BIMs (see (b))}} - \underbrace{\gamma \cdot \frac{\sum_{o \in O} d_o}{T}}_{\text{break lengths (see (c))}} + \underbrace{\delta \cdot \left( \sum_{i \in I, o \in O} u_{io} - a \right)}_{\text{changing pattern (see (d))}} \quad (1)$$

with the constraints

assignment

$$\sum_{o \in O} y_{io} = 1 \quad \forall i \in I \quad (2)$$

$$s_i + p_i \leq T \quad \forall i \in I \quad (3)$$

job sequence

$$s_j - s_i \leq x_{ij} \cdot M \quad \forall i, j \in I : i \neq j \quad (4)$$

$$x_{ij} = 1 - x_{ji} \quad \forall i, j \in I : i \neq j \quad (5)$$

$$s_i + p_i \leq s_j + (1 - z_{ij}) \cdot M \quad \forall i, j \in I : i \neq j \quad (6)$$

$$s_i + p_i + \varepsilon \geq s_j - z_{ij} \cdot M \quad \forall i, j \in I : i \neq j \quad (7)$$

break length

$$s_i + p_i + d_o \leq s_j + (3 - x_{ij} - y_{io} - y_{jo}) \cdot M \quad \forall o \in O, i, j \in I : i \neq j \quad (8)$$

$$n_o \leq \frac{\sum_j y_{jo}}{2} \quad \forall o \in O \quad (9)$$

$$d_o \leq n_o \cdot M \quad \forall o \in O \quad (10)$$

overlap

$$s_i + p_i - s_j \leq ov_{ij} + (1 - x_{ij} + w_{ij}) \cdot M \quad \forall i, j \in I : i \neq j \quad (11)$$

$$p_j \leq ov_{ij} + (1 - w_{ij}) \cdot M \quad \forall i, j \in I : i \neq j \quad (12)$$

BII

$$s_i + p_i - (s_j + p_j) \leq b_j + \left( x_{ij} \cdot (|I| + 1) + \left( \sum_{k \in I: k \neq i, j} x_{ki} - x_{kj} \right) - 1 \right) \cdot M$$

$$\forall i, j \in I : i \neq j$$

$$s_i + p_i - (s_j + p_j) \geq b_j \quad (13)$$

$$- \left( x_{ij} \cdot (|I| + 1) + \left( \sum_{k \in I: k \neq i, j} x_{ki} - x_{kj} \right) - 1 \right) \cdot M$$

$$\forall i, j \in I : i \neq j \tag{14}$$

$$b_i \leq T - s_i - p_i + M \cdot \sum_{j \in I: j \neq i} x_{ij} \quad \forall i \in I \tag{15}$$

$$b_i \geq T - s_i - p_i - M \cdot \sum_{j \in I: j \neq i} x_{ij} \quad \forall i \in I \tag{16}$$

$$b_i \leq \bar{b} \quad \forall i \in I \tag{17}$$

$$b_i \geq \underline{b} - (1 - c_i) \cdot M \quad \forall i \in I \tag{18}$$

$$s_i + b_i \leq c_i \cdot M \quad \forall i \in I \tag{19}$$

changing pattern

$$v_{ijo} \geq x_{ji} + y_{jo} - 1 \quad \forall o \in O, i, j \in I : i \neq j \tag{20}$$

$$v_{ijo} \leq (x_{ji} + y_{jo})/2 \quad \forall o \in O, i, j \in I : i \neq j \tag{21}$$

$$v_{iio} = y_{io} \quad \forall i \in I, o \in O \tag{22}$$

$$u_{io} \geq \left( \sum_{k \in I: k \neq i, j} x_{ki} - x_{kj} \right) - M \cdot (1 - y_{jo})$$

$$- M \cdot \left( \sum_{k \in I: k \neq i} v_{iko} - \sum_{k \in I} v_{jko} \right) - M \cdot |I| \cdot x_{ij}$$

$$\forall o \in O, i, j \in I : i \neq j \tag{23}$$

$$u_{io} \leq \left( \sum_{k \in I: k \neq i, j} x_{ki} - x_{kj} \right) + M \cdot (1 - y_{jo})$$

$$+ M \cdot \left( \sum_{k \in I: k \neq i} v_{iko} - \sum_{k \in I} v_{jko} \right) + M \cdot |I| \cdot x_{ij}$$

$$\forall o \in O, i, j \in I : i \neq j \tag{24}$$

$$a \leq u_{io}/2 + (1 - y_{io}) \cdot M + \left( |I| - 1 - \sum_{j \in I: j \neq i} x_{ji} \right) \cdot M$$

$$\forall i \in I, o \in O \tag{25}$$

range of decision variables

$$a, \bar{b}, \underline{b} \geq 0 \tag{26}$$

$$b_i, s_i \geq 0, c_i \in \{0, 1\} \quad \forall i \in I \tag{27}$$

$$d_o \geq 0, n_o \in \{0, 1\} \quad \forall o \in O \tag{28}$$

$$ov_{ij} \geq 0, w_{ij}, x_{ij}, z_{ij} \in \{0, 1\} \quad \forall i, j \in I : i \neq j \tag{29}$$

$$u_{io} \geq 0, y_{io} \in \{0, 1\} \quad \forall i \in I, o \in O \tag{30}$$

$$v_{ijo} \in \{0, 1\} \quad \forall i, j \in I, o \in O \tag{31}$$

The objective function optimizes our four goals stated in Sect. 3.2. The letters in brackets refer to the letters of the optimization goals. In particular, the objective minimizes the overlap between jobs (11)–(12); compare objective (a) and the deviation between the longest and shortest break-in-interval (13)–(19); compare objective (b)). Moreover, we maximize the shortest break length on each resource (8)–(10); compare objective (c) and optimize the rotation of jobs amongst resources (changing pattern; (20)–(25); compare objective (d)). (2) makes sure that each job is scheduled on exactly one resource. Because of (3) it is not possible that the sum of processing times on one resource is bigger than the available time  $T$ .

Constraints (4)–(7) determine the job sequence. Due to (4),  $x_{ij} = 1$  if job  $j$  starts after job  $i$ . (5) makes sure that there is a unique sequence even if more than one job starts at the same point in time. Thanks to Constraints (7)  $z_{ij} = 1$  if job  $i$  is finished before job  $j$  starts. Due to  $\varepsilon$ , the small positive number,  $z_{ij} = 1$  even if job  $i$  is finished at the point in time job  $j$  starts. Constraints (6) make sure that  $z_{ij}$  is zero if job  $i$  is not finished when job  $j$  starts.

Constraints (8)–(10) determine the shortest break length on all resources. As the objective tries to maximize  $d_o$ , (8) makes sure that  $d_o$  equals the smallest break on resource  $o$ , i.e. the smallest time between the end of a job  $i$  ( $s_i + p_i$ ) and the start of the following job  $j$ . Due to  $(3 - x_{ij} - y_{io} - y_{jo}) \cdot M$ , only the following jobs on the same resource are taken into account. Constraints (9) and (10) are only necessary if only one job is assigned to resource  $o$ . As in this case  $d_o$  is not bounded by (8), (10) fixes it to zero.

The next block of constraints determines the overlap-variables  $ov_{ij}$ , which indicate the amount of time jobs  $i$  and  $j$  run parallel. Constraints (11) compute the overlap if job  $j$  is a successor of job  $i$ , and job  $i$  ends before  $j$  does ( $w_{ij} = 0$ ). If job  $j$  ended before job  $i$ , (11) would lead to a wrong overlap. Therefore,  $w_{ij}$  is introduced and the case in which job  $i$  starts before and ends after job  $j$  is considered in Constraints (12). Note that an overlap of more jobs is worse than of fewer. If there is an overlap of two jobs, (11) and (12) count the duration when the second job starts. If there is an overlap of three jobs, the overlap between the first two will be counted when the second job starts. Furthermore, the overlap between the third and the first, and the third and the second job is counted when the third job starts. So the overlap between all three of them is counted three times.

Break-in-intervals are considered in Constraints (13)–(19). Constraints (13) and (14) fix the BII, i.e. the time between the end time of job  $j$  and the end time of the job  $i$  which starts next to  $b_j$ , for all jobs which have a successor. Constraints (15) and (16) do the same for the last job on each resource. In this case, the break-in-interval is the time until the end of the planning horizon  $T$ , as we assume that emergency jobs can enter any resource at the end of the planning

horizon. Constraints (17) and (18) determine the maximal and minimal break-in-intervals  $\bar{b}$  and  $\underline{b}$ . For the computation of  $\underline{b}$  only one job which starts at the beginning of the planning horizon is counted ( $c_i = 1$  for all other jobs starting at the beginning of the planning horizon due to (19)), as  $\underline{b}$  would be zero otherwise.

In the following, the changing pattern is defined. For this, we introduce  $v_{ijo}$  in Constraints (20)–(22) which is 1 if job  $j$  starts before job  $i$  and job  $j$  is assigned to resource  $o$ . We need  $v_{ijo}$  for the big  $M$  in Constraints (23) and (24), in which the number of jobs which started between the start of the latest preceding job on resource  $o$  and the start of job  $i$  is counted ( $u_{io}$ ). The big  $M$  part with  $v_{ijo}$  makes sure that this last preceding job on resource  $o$  is considered. Constraints (25) make sure that the last job  $i$  is assigned to the resource for which  $u_{io}$  is maximal, i.e. for which the last preceding job is the longest time ago. The factor  $1/2$  is important, as  $i$  is also counted in the sum of the third part of the objective (*changing pattern*). Without the factor, the model has no pressure to assign  $i$  to the desired resource.

We need  $\mathcal{O}$  dummy jobs, one at time zero on each resource to initialize the changing pattern. Constraints (6), (7), and (18) do not apply for them. Constraints (19) make sure that Constraints (18) are not binding in this case. Furthermore, we cannot count any of them in Constraints (8). Otherwise we would have a break at time zero on each resource (see Theorem 1). Finally, Constraints (26)–(31) are the non-negativity and binary constraints.

### 5 Heuristic scheduling approach

The developed *MIP* model contains a significant number of binary variables and will likely be restricted to solving instances of small size only. Thus, we develop a construction heuristic in this section that aims to approximate an ideal schedule structure based on the insights of Sect. 3.

Consider the schedule in Fig. 4. Here, 7 jobs have been scheduled on three resources such that all of the four objectives described above are satisfied ideally, i.e. breaks are evenly distributed over time ((b)) and resources ((a)), the lengths of breaks between any two consecutive jobs are identical ((c)), and the jobs rotate from one resource to the next in the sequence (2, 3, 1) ((d)). We will use the structure of the above schedule as a blueprint to construct approximate schedules of similar quality. For this purpose, we will interpret the ideally distributed capacity utilization in the schedule of Fig. 4 as slots, to which a set of jobs of a given problem instance needs to be assigned. Of course, it will typically not be possible to assign jobs perfectly to the start and end times of the slots, but in getting as close as possible, we will produce schedules that maintain much of its favourable structure. In what follows, we will assume w.l.o.g. that there are at least

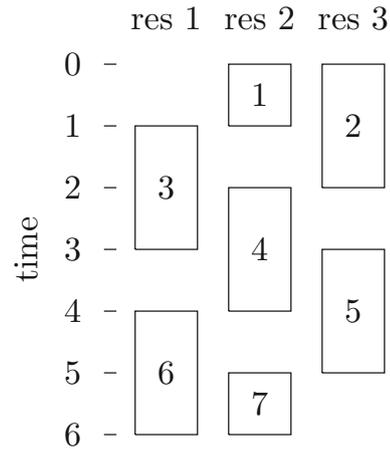


Fig. 4 Example schedule for the heuristic approach

as many jobs as resources, since any excess resource will not need to be considered for a job assignment.

We will first characterize the distribution of slots in the ideal schedule and then explain how jobs of a given planning instance are assigned to slots. At the core of the schedule, there is a set  $B$  of slots of the same length which are successively assigned to resources rotating from one to the next (jobs 3–5 in Fig. 4). The exact number of slots in  $B$  depends on the number of complete rotations  $R$  (here a single rotation) and the number of resources ( $|B| = R \cdot \mathcal{O}$ ). In addition to that, there is an incomplete rotation at the beginning of the schedule of slots in  $A$  with  $|A| \leq \mathcal{O}$  that have varying lengths and make use of any underutilized resources at the beginning of the schedule (here jobs 1 and 2) and another incomplete rotation at the end of the schedule of slots in  $C$  with  $|C| \leq \mathcal{O}$  that also have different lengths which utilize the remaining capacity at the end of the planning horizon  $T$  (here jobs 6 and 7).

The construction heuristic works in three phases. In the first phase, we determine the ideal structure of the schedule, that is we determine the number and start and end times of slots on each resource. In the second phase, jobs are assigned to the sets of slots, and in the third phase the actual start times of jobs on resources are determined. We begin by determining the cardinality of each of the slot sets. Given that each resource has  $T$  units of productive time, an even distribution of breaks over resources seeks to keep approximately  $\mathcal{O} - \frac{\sum_i p_i}{T}$  resources free at any point in time. It follows that set  $A$  of the first incomplete rotation should ideally contain

$$|A| := \left\lfloor \frac{\sum_i p_i}{T} \right\rfloor \tag{32}$$

slots, where  $\lfloor \cdot \rfloor$  rounds to the nearest integer. As a consequence, the number of complete rotations over all resources equals

$$R := \left\lfloor \frac{|I| - |A|}{\mathcal{O}} \right\rfloor \tag{33}$$

and thus  $|B| = R \cdot \mathcal{O}$ . Any remaining slots are assigned to set  $C$  as the final incomplete rotation

$$|C| := |I| - |A| - |B|. \tag{34}$$

There are several possible ways to determine lengths and start times of slots which utilize a resource in an ideal schedule which can be characterized by a divisor  $D$  that denotes the number of slots over which utilized and idle times are to be distributed. Once  $D$  is determined, the lengths of the slots  $sl = \frac{\sum_i p_i}{D}$  and breaks  $bl = \frac{\mathcal{O} \cdot T - \sum_i p_i}{D}$  are determined by evenly distributing the utilized and idle time over all slots. Finally, the slots should be evenly distributed over the time horizon, so we set the according rate  $sr$  to  $\frac{T}{D}$ . One natural way of choosing  $D$  is to count the number of breaks that are to be scheduled on all resources as  $D = (R + 1) \cdot \mathcal{O}$ , one initial break on each resource and an additional one after each slot in a rotation. This is the case in the example Fig. 4, where  $D = (1 + 1) \cdot 3 = 6$  and thus  $sl = \frac{12}{6} = 2$ ,  $bl = \frac{6}{6} = 1$  and  $sr = \frac{6}{6} = 1$ . However, whenever the actual job lengths deviate from the lengths of the ideal slots, better approximations might be generated by varying  $D$ .

With the help of these three concepts, we can determine the ideal start times  $s_j^*$  and end times  $e_j^*$  of the slots. Assuming that slots are numbered in ascending order, the first  $|A|$  slots all have start times of  $s_j^* = 0$  for  $j = 1, \dots, |A|$ . The start times of the next  $|B| + |C|$  slots are spaced over time by the ideal rate such that they result to  $s_{|A|+j}^* = sr \cdot j$  for  $j = 1, \dots, |B| + |C|$ . Ideal end times of the first  $|A|$  slots depend on the start times of the next slot on the same resource and the ideal length of the break between any two slots. It thus follows that  $e_j^* = sr \cdot (\mathcal{O} - |A| + j) - bl$  for  $j = 1, \dots, |A|$ . End times for the remaining slots depend on the ideal slot length and the end of the planning horizon  $e_{|A|+j}^* = \min\{s_{|A|+j}^* + sl, T\}$  for  $j = 1, \dots, |B| + |C|$ .

In the next phase, the jobs of the planning instance are grouped into subsets according to the slots they are later assigned to. Let  $I^A, I^B$ , and  $I^C$  refer to the job sets assigned to the corresponding sets of slots. Since the slots in  $A$  and  $C$  will typically be shorter, jobs with the lowest processing times are interchangeably assigned to sets  $I^A$  and  $I^C$ , respectively. The remaining jobs are assigned to  $I^B$  as described by the following pseudocode:

**Algorithm 1**

```

1: for j = 1 to max(|A|, |C|) do
2:   if |IA| < |A| then
3:     Remove the job i* with the smallest processing time
       from I
4:     Add i* to IA
5:   end if
6:   if |IC| < |C| then
7:     Remove the job i* with the smallest processing time
       from I
8:     Add i* to IC
9:   end if
10: end for
11: IB = I
    
```

Consider the example again with 7 jobs with  $p_1 = p_7 = 1$  and  $p_i = 2 \forall i \in \{2, \dots, 6\}$  in Fig. 4. Since  $\sum_i p_i = 12$ , two jobs are assigned to the two incomplete rotations at the beginning and the end ( $|A| = |C| = 2$ ). So, we have two runs through the ‘for’-loop (lines 1–10). In the first, one of jobs 1 and 7 is assigned to  $I^A$  and the other to  $I^C$ . As all remaining jobs have the same length, one of them is assigned to  $I^A$  and one to  $I^C$  in the second run of the ‘for’-loop. The remaining jobs are assigned to  $I^B$  (line 11).

Finally, for each job  $i$  the start time  $s_i$  and the assignment variables  $y_{i,o}$  of job  $i$  to resource  $o$  are determined. Generally speaking, we seek to start jobs at the ideal start time of the slot they are assigned to. However, the schedule is adapted whenever some prior job either fell below or exceeded their respective slot in order to compensate for the additional or reduced length of the break in the schedule. Notice in Fig. 4 that if job 1 were to finish prior to the end of its slot, this could be compensated by letting job 3 start earlier by the same amount. In particular, if job 3 were itself longer than its ideal slot length, it might just finish right with the end of its slot and the rest of the schedule would be unaffected. We will make use of this insight by tracking for any resource  $o$  the deviation  $dev_o$  which its last assigned job had from the ideal end date and then adjusting the next job that is assigned to resource  $o' = 1 + (o + |A| - 1) \bmod \mathcal{O}$ , where mod is the modulo division. This ensures that the two resources affected by this adjustment are spaced by  $|o - o'| = \mathcal{O} - |A|$ , i.e. the number of resources that have a scheduled break at any time in the ideal schedule. The whole procedure is given by the following pseudocode:

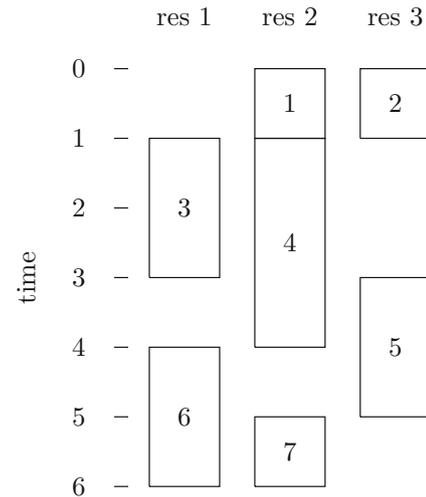
**Algorithm 2**

```

1: for  $j = 1$  to  $|A|$  do
2:   Remove the job  $i^*$  with the smallest processing time
   from  $I^A$ 
3:    $s_{i^*} = s_j^*$ 
4:    $o = \mathcal{O} - |A| + j$ 
5:    $y_{i^*,o} = 1$ 
6:    $dev_o = p_{i^*} - e_j^*$ 
7:    $end_o = p_{i^*}$ 
8: end for
9: for  $j = 1$  to  $R$  do
10:  for  $o = 1$  to  $\mathcal{O}$  do
11:    $j' = (j - 1) \cdot \mathcal{O} + o$ 
12:    $o' = 1 + (o + \mathcal{O} - |A| - 1) \bmod \mathcal{O}$ 
13:    $s' = s_{|A|+j'}^* + dev_{o'}$ 
14:   if  $s' < end_o$  then
15:      $s' = end_o$ 
16:   end if
17:   Remove the job  $i^*$  that minimizes  $|e_{j'}^* - s' - p_{i^*}|$ 
   from  $I^B$ 
18:    $s_{i^*} = s'$ 
19:    $y_{i^*,o} = 1$ 
20:    $dev_o = s_{i^*} + p_{i^*} - e_j^*$ 
21:    $end_o = s_{i^*} + p_{i^*}$ 
22: end for
23: end for
24: for  $j = 1$  to  $|C|$  do
25:   $o' = 1 + (j + \mathcal{O} - |A| - 1) \bmod \mathcal{O}$ 
26:   $s' = s_{|A|+|B|+j}^* + dev_{o'}$ 
27:  if  $s' < end_j$  then
28:     $s' = end_j$ 
29:  end if
30:  Remove the job  $i^*$  with the largest processing time
   from  $I^C$ 
31:   $s_{i^*} = s'$ 
32:   $y_{i^*,j} = 1$ 
33:   $end_j = s_{i^*} + p_{i^*}$ 
34: end for
35: for  $o = 1$  to  $\mathcal{O}$  do
36:  if  $end_o > T$  then
37:    Reschedule the last job on resource  $o$  and, if neces-
    sary, its predecessors on resource  $o$ , to an earlier
    point in time until the time horizon is met
38:  end if
39: end for

```

Jobs in  $I^A$  are considered in ascending order of their processing times and start at their ideal slot lengths, i.e. the beginning of the planning horizon (lines 2–3). Since  $\mathcal{O} - |A|$  resources are supposed to be free at any time, job  $j$  is assigned to resource  $\mathcal{O} - |A| + j$  (lines 4–5). For each resource  $o$ , the deviation from the ideal end date  $dev_o$  and the actual point



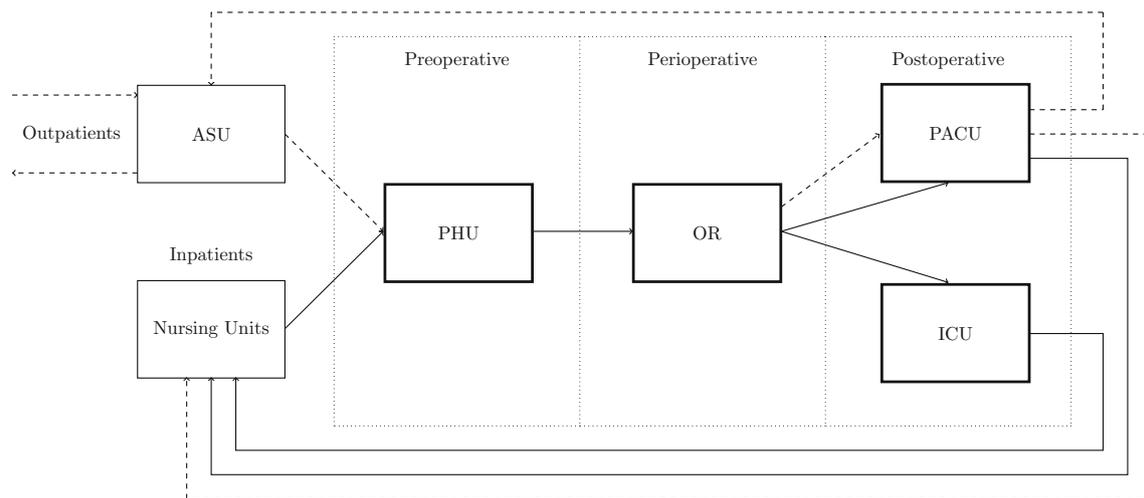
**Fig. 5** Alternative example for the heuristic approach

in time when  $o$  is free again  $end_o$  is stored (lines 6–7). The procedure then moves to schedule the jobs in  $I^B$ . First, the start time of the job  $s'$  is set to the ideal start time of its slot (line 13) adjusted by the deviation incurred by the job that was scheduled  $|A|$  resources before (line 12) or to  $end_o$  in case resource  $o$  is not free (lines 14–16). Then, the job in  $I^B$  that best fits within the adjusted slot length is identified and scheduled on the current resource (lines 17–21). Finally, the jobs in  $I^C$  are scheduled in the same fashion (lines 24–34). However, since the slots in  $C$  will get shorter and shorter, the jobs in  $I^C$  are scheduled in descending order of their processing times (line 30), rather than on a best fit basis.

With  $I^A = \{1, 2\}$ ,  $I^B = \{3, 4, 5\}$ , and  $I^C = \{6, 7\}$  in the example in Fig. 4, jobs 1 and 2 are assigned to resources  $\mathcal{O} - |A| + 1 = 3 - 2 + 1 = 2$  and  $\mathcal{O} - |A| + 2 = 3$  in lines 1–8 of the algorithm. Afterwards, lines 9–23 assign jobs 3, 4, and 5 in any order to the three resources, as all of them have the same length.

In this example, all jobs fit perfectly to their corresponding slots, so no adjustments to start or end times were necessary. In order to see how start and end times are adapted, consider the variation of the example with  $p_2 = 1$  and  $p_4 = 3$  in Fig. 5. There, job 2 would also be assigned to  $I^A$ ; however, its scheduling on resource 3 would lead to a deviation of  $dev_3 = -1$ , since it would end too early for its slot. This is considered when computing the starting time of job 4 (lines 11–13). With  $j' = (j - 1) \cdot \mathcal{O} + o = (1 - 1) \cdot 3 + 2 = 2$ ,  $o' = 1 + (o + \mathcal{O} - |A| - 1) \bmod \mathcal{O} = 1 + (2 + 3 - 2 - 1) \bmod 3 = 3$  and  $s' = s_{|A|+j'}^* + dev_{o'} = s_4 + dev_3 = 1$ . In this way, the starting time of job 4 ( $s_4$ ) is corrected by the amount of time job 2 finishes too early and the planned breaking interval is in effect shifted from resource 2 to resource 3, in order to account for the different job lengths.

Notice that the construction heuristic cannot guarantee that all jobs which are assigned to slots in this way end before



**Fig. 6** Scheme of a surgical centre (Pham and Klinkert, 2008)

$T$  on all resources. This is not surprising, since ensuring feasibility would require the solution of a bin packing problem, which is well known to be NP-hard in the strong sense. In the case that a job exceeds  $T$  on a resource, we thus use the following simple repair strategy: Beginning with the last job on the resource, reschedule all jobs successively to an earlier point in time until the time horizon is met (lines 35–39). This repair strategy will tend to lead to an overutilization of resources at the end of the planning horizon, the more elective surgeries have to be pushed back in this manner. It might thus make sense to choose a higher divisor  $D$  to motivate a more dense assignment of jobs at the beginning of the planning horizon and thus potentially avoiding repair operations. Preliminary computational experiments have indicated that  $D = |I|$  offers a good compromise between these two effects and is thus chosen for the following experimental evaluation.

## 6 Computational study

In this section, we employ the deterministic scheduling approaches on an example case from the field of operating room scheduling and compare the results to other scheduling policies from the literature. We will first give a brief introduction to the planning environment.

### 6.1 Description of planning environment

Figure 6 shows a typical set-up of a surgical operations centre. Elective patients arrive either from outside the hospital as ambulatory patients just for their surgery or they come from a nursing unit, but typically every scheduled patient arrives at a fixed appointment in the surgical centre. In the first case, the incoming patient registers in the ambulatory surgical unit (ASU) and then moves into the preoperative holding unit (PHU), where the patient is prepared for the

surgery. Inpatients are brought directly into the PHU. From there all patients enter the OR, where the surgery takes place. As soon as the surgery is finished, the patient is either brought to the post-anaesthesia care unit (PACU), where the patient recovers from anaesthesia, or to the intensive care unit (ICU) if the status of the patient is critical. Inpatients are brought back to their nursing unit after finishing in PACU, while outpatients either return to ASU and leave the hospital or are moved to a nursing unit for post-operative care.

In the following, we will focus on the perioperative area and consider the scheduling of elective surgeries to operating rooms. We assume that all elective surgeries have already been assigned to a given day in a prior planning step and thus restrict our analysis to the short-term scheduling of daily surgeries to operating rooms. In order to test the performance of our approaches, of the *MIP* model and the construction heuristic (referred to as *breaks*), we compare them with alternative proposals from the literature. A standard policy is to reserve an OR for emergencies exclusively and use only the remaining ORs for elective scheduling. We will refer to this policy as *exclusive* in the following (see van Essen et al., 2012) and use it as a benchmark for calculating capacity utilization. Wullink et al. (2007) propose to schedule all elective surgeries in the first part of the day using all ORs and keep the remainder of their capacity reserved for emergencies. In order to allow for break-ins during the first part, they set BIMs between elective surgeries but do not consider breaks. van Essen et al. (2012) develop several formal methods to evenly distribute start times of surgeries. We implemented their fixed goal value heuristic which demonstrated some of the strongest performances in order to schedule break-in-moments. Since the heuristic requires a prior assignment of surgeries to ORs, jobs are first ordered in descending order of their processing time and then successively assigned to the OR with the current lowest sum of surgery times. We refer to this approach as *BIM*. Finally, we also test a combina-

tion of the two alternative proposals by reserving an OR for emergencies, but allowing break-ins into the elective schedule at evenly distributed BIMs whenever the exclusive OR is occupied (*BIM + ex*).

### 6.2 Parameters of the simulation environment

We simulate two sets of instances, a small test set and a large test set with varying numbers of ORs and elective surgeries. In the small test set, either two or three ORs are considered, one of which is reserved for emergencies in the benchmark policy. Elective surgeries are varied between 4 and 16 (4, 6, and 8 per non-reserved OR) as given in Table 1.

The length of the surgery day was set to eight hours (480 minutes) and the total sum of all surgery durations fixed at  $(C - 1) \cdot 480$ , so that capacity utilization of non-reserved ORs is set to 100%. Processing times of elective surgeries are determined by a log-normal distribution (compare Strum et al., 2000 and van Essen et al., 2012) with a shifting mean set to the remaining surgery time divided by the remaining number of surgeries with a standard deviation of 10.

In the large test set, up to 15 ORs are scheduled with 4, 6, and 8 elective surgeries for any non-reserved OR as given in Table 2. Capacity utilization of non-reserved ORs was further varied in steps of 80%, 90%, and 100% to account for situations when the capacity of non-reserved ORs is not fully utilized by elective surgeries.

For each combination of parameters, 60 scheduling instances are created and solved by the scheduling policies. The small set was solved by all approaches, including the *MIP* model which was implemented in GAMS and solved by CPLEX with a time limit of 3600s (= 1 hour). The large set was solved by the other four policies only, which were implemented in *VB.net*. After the instances were solved, each solution was evaluated in 500 simulation runs to compute waiting times. For the simulation, arrival times of emergencies followed a Poisson process with an expected value of 4/480, so that four emergency patients per day are expected. Emergency surgery times were set by a log-normal distri-

bution (compare Wullink et al., 2007, de Bruin et al., 2010) with a mean of 90 minutes and a standard deviation of 10.

The simulation then proceeds as follows: Whenever an emergency enters the system, the next possible break-in point in the elective schedule is determined (i.e. the next point in time an OR is free) and the emergency is assigned to this OR. If the assigned emergency interferes with one or more scheduled elective surgeries, then these electives are marked as postponed. In practice, there are several ways to react to such an interference. For instance, the elective surgery could simply be postponed until the OR is free again or assigned to another OR in the meantime. Alternatively, the elective could be moved to an overtime period at the end of the shift or be cancelled completely for the day and be rescheduled at a later date. In order not to rule out particular reaction schemes and still test the performance of the schedule structures created by the scheduling procedures, we simply count postponed electives, but do not allow further emergencies to make use of any additional free time which might result from a cancellation. This way it is also ensured that the capacity utilization remains constant over all procedures, which seems a prerequisite for a fair comparison.

### 6.3 Results of the simulation

Preliminary tests showed that *MIP* could only solve instances up to three ORs reliably, so that the performance of the approach is investigated for the small test set only. We set  $\alpha = \beta = \gamma = \delta = 1$ . Hence, all parts of the objective have the same weight. Table 3 shows the average waiting times over all instances with two and three ORs and over all instances with the given number of surgeries.

As can be seen, the benchmark policy *exclusive* leads to the highest average waiting time of over 45 minutes. Since the exclusive reservation decouples elective and emergencies, the results are further independent of the number of ORs and elective surgeries. In line with the results of Wullink et al. (2007), waiting times can be reduced significantly by making use of all ORs and evenly distributing BIMs over the surgery day (*BIM*). Yet, the combination of both approaches *BIM +*

**Table 1** Tested combinations of number of ORs (#OR) and number of elective surgeries (#elec) for small test set

	1	2	3	4	5	6
#OR	2	2	2	3	3	3
#elec	4	6	8	8	12	16

**Table 2** Tested combinations of number of ORs and number of elective surgeries for large test set

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
#OR	2	2	2	3	3	3	5	5	5	10	10	10	15	15	15
#elec	4	6	8	8	12	16	16	24	32	36	54	72	56	84	112

**Table 3** Expected waiting times over small test set

#OR	2	3	#elec	4	6	8	12	16	avg
<i>exclusive</i>	45.22			45.22					45.22
<i>BIM</i>	30.58	21.59		41.51	28.89	25.63	21.25	13.60	26.09
<i>BIM + ex</i>	18.78	12.38		23.53	20.37	16.79	11.81	8.95	16.37
<i>MIP</i>	20.95	9.84		26.06	22.46	15.62	6.95	5.63	15.39
<i>breaks</i>	20.29	8.87		25.15	21.21	14.88	6.71	4.62	14.57

**Table 4** Results of the simulation experiment of large test set

Util	#OR	#elec	<i>breaks</i>		<i>BIM</i>		<i>exclusive</i>		<i>BIM + ex</i>	
			Avg. wait (max)	%postp	Avg. wait (max)	#postp	Avg. wait (max)	#postp	Avg. wait (max)	#postp
0.8	2	4	20.20 (38.03)	2.27	26.89 (50.49)	2.27	45.22 (72.36)	0	18.59 (35.40)	1.43
0.8	2	6	15.15 (29.56)	3.23	20.11 (37.94)	3.03			15.28 (29.87)	2.09
0.8	2	8	12.08 (23.41)	4.04	15.37 (29.94)	3.54			13.56 (26.94)	2.63
0.8	3	8	9.85 (22.61)	3.07	18.29 (40.01)	3.55			11.14 (24.92)	1.83
0.8	3	12	4.84 (11.46)	4.03	13.10 (27.53)	4.80			7.87 (17.74)	2.73
0.8	3	16	3.05 (7.90)	4.30	8.52 (18.24)	5.57			6.04 (14.18)	3.28
0.8	5	16	1.63 (4.71)	3.61	17.61 (37.64)	4.97			7.89 (19.84)	2.09
0.8	5	24	0.47 (1.50)	3.96	8.24 (18.53)	6.10			4.36 (11.22)	3.08
0.8	5	32	0.49 (1.54)	5.11	5.31 (11.69)	7.14			2.96 (7.79)	3.64
0.8	10	36	0.18 (0.63)	3.49	13.70 (32.38)	5.33			6.14 (16.43)	2.27
0.8	10	54	0.04 (0.16)	4.53	6.20 (14.57)	7.34			2.88 (8.16)	3.38
0.8	10	72	0.03 (0.10)	6.21	3.64 (8.46)	8.29			1.62 (4.67)	4.08
0.8	15	56	0.24 (0.89)	3.49	12.15 (29.83)	5.40			5.56 (15.21)	2.38
0.8	15	84	0.03 (0.13)	4.77	5.31 (13.39)	7.54			2.31 (6.85)	3.56
0.8	15	112	0.01 (0.02)	6.51	2.66 (6.63)	8.60			1.16 (3.49)	4.29
0.9	2	4	23.42 (43.29)	2.47	32.74 (58.78)	2.53			20.65 (38.79)	1.48
0.9	2	6	18.07 (34.20)	3.58	24.42 (44.37)	3.33			17.99 (34.16)	2.26
0.9	2	8	12.96 (24.71)	4.17	18.21 (34.13)	3.88			15.02 (29.52)	2.74
0.9	3	8	12.19 (27.01)	3.27	23.41 (48.49)	3.85			13.72 (29.34)	1.94
0.9	3	12	5.71 (13.26)	4.23	17.28 (34.47)	5.32			9.80 (21.32)	2.94
0.9	3	16	3.77 (9.39)	4.43	10.81 (22.07)	6.06			7.51 (16.84)	3.45
0.9	5	16	9.01 (23.05)	3.49	23.67 (47.39)	5.23			10.24 (24.32)	2.17
0.9	5	24	3.04 (8.80)	4.44	11.17 (23.91)	6.68			5.87 (14.59)	3.32
0.9	5	32	1.39 (4.20)	5.63	6.97 (14.52)	7.72			3.74 (9.47)	3.92
0.9	10	36	4.00 (12.03)	4.08	18.83 (41.71)	5.58			8.40 (21.27)	2.37
0.9	10	54	0.60 (2.08)	4.82	8.39 (18.82)	8.06			3.77 (10.28)	3.63
0.9	10	72	0.24 (0.83)	6.73	4.95 (10.98)	8.99			2.23 (6.20)	4.28
0.9	15	56	5.96 (17.16)	4.21	18.01 (40.64)	5.65			7.69 (20.00)	2.44
0.9	15	84	0.74 (2.53)	4.92	7.50 (17.91)	8.29			3.26 (9.18)	3.76
0.9	15	112	0.17 (0.64)	6.90	3.81 (9.14)	9.25			1.67 (4.89)	4.55
1.0	2	4	25.15 (45.98)	2.71	41.51 (71.95)	2.77			23.53 (43.42)	1.36
1.0	2	6	21.21 (38.52)	4.03	28.89 (51.28)	3.58			20.37 (37.74)	2.20
1.0	2	8	14.51 (27.26)	4.47	21.34 (39.13)	4.23			17.19 (33.06)	2.81
1.0	3	8	15.27 (32.07)	3.54	29.92 (58.41)	4.25			16.39 (34.25)	2.01
1.0	3	12	6.71 (15.34)	4.40	21.25 (40.34)	5.66			11.82 (25.04)	3.00
1.0	3	16	4.62 (11.25)	4.68	13.60 (26.40)	6.61			8.95 (19.34)	3.68
1.0	5	16	10.73 (26.43)	3.89	30.09 (57.05)	5.69			13.07 (29.43)	2.28
1.0	5	24	4.43 (11.85)	4.75	13.84 (28.02)	7.06			7.15 (17.11)	3.45

Table 4 continued

Util	#OR	#elec	<i>breaks</i>		<i>BIM</i>		<i>exclusive</i>			<i>BIM + ex</i>	
			Avg. wait (max)	%postp	Avg. wait (max)	#postp	Avg. wait (max)	#postp	Avg. wait (max)	#postp	
1.0	5	32	2.34 (6.65)	6.19	8.90 (17.73)	8.35				5.02 (12.24)	4.15
1.0	10	36	9.77 (24.92)	4.14	25.25 (52.38)	6.11				10.96 (26.29)	2.49
1.0	10	54	3.05 (8.91)	5.40	11.70 (24.79)	8.59				5.10 (13.36)	3.75
1.0	10	72	1.62 (4.99)	7.40	6.45 (13.62)	9.77				2.91 (7.87)	4.63
1.0	15	56	9.62 (24.47)	4.37	23.24 (49.84)	6.20				9.97 (24.67)	2.62
1.0	15	84	3.18 (9.32)	5.70	10.14 (22.88)	8.79				4.43 (12.00)	3.90
1.0	15	112	1.38 (4.33)	7.66	5.05 (11.59)	10.00				2.23 (6.34)	4.88
			6.74 (14.63)	4.52	15.52 (31.33)	6.03	45.22 (72.36)	0		8.84 (19.45)	3.00

*ex*, where a room is exclusively reserved and yet break-ins are allowed in the elective schedule, considerably improves performance even further, since the free capacity is more evenly distributed over time in this case. The two scheduling policies developed in this work lead to a further decrease in waiting times of about 2 minutes compared to *BIM + ex* on average over all considered instances. However, they perform worse than *BIM + ex* for the smallest instance sets with two ORs and 4 to 6 electives, but significantly better once the number of ORs and elective surgeries is increased. This is to be expected, since few surgeries allow only few break-in-moments, so that an exclusive OR for emergencies only will have a bigger advantage. Both *MIP* and *breaks* perform similarly and dominate *BIM*. Interestingly, *MIP* is even slightly beaten by the simple construction heuristic because *MIP* could not solve all instances to optimality and might not be parameterized optimally.

In order to get a better insight into the performance, all approaches, except for *MIP*, have further been tested on the large test set. Table 4 gives an overview over all results reporting the average waiting time of emergencies and the maximum waiting times per simulation run averaged over all runs. Furthermore, it displays the average number of elective surgeries that have to be postponed due to a break-in of emergencies. Over all instances, *breaks* clearly outperforms the benchmark policy and *BIM* at least with respect to average and maximum waiting times. The exclusive reservation has the advantage that the elective schedule is never disrupted, while break-ins regularly lead to a postponement of electives in the other cases. When comparing *breaks* with *BIM*, we can see that the latter is dominated in terms of waiting times and number of postponed electives. The combination of an exclusively reserved OR with additional break-ins on average performs worse with respect to waiting times when compared to *breaks*, yet the performance trade-off is much more complex, also because the lower average waiting times generated by *breaks* will be weighed against the lower number of postponed electives in the case for *BIM + ex*. On average

over all instances, *breaks* gains approximately 1.5 minutes in lower waiting times at the cost of an additional elective surgery having to be postponed. Yet, the performance differences vary considerably with respect to capacity utilization, number of ORs and electives.

In tendency, the lower the number of elective surgeries and the higher the capacity utilization the better the results of *BIM + ex* when compared to *breaks*. This makes intuitive sense, since *breaks* is designed to spread out the idle time of ORs and thus the more idle time there is to work with, the better it will perform in comparison. Likewise, a low number of elective surgeries will only allow few degrees of freedom for break-ins and thus the exclusive OR provides more availability. For the same reason, the performance of all approaches (except *exclusive*) tends to improve with an increasing number of elective surgeries per OR. Interestingly, while the two *BIM* approaches also benefit from an increased number of ORs, the performance of *breaks* is improving at first, but then in some instances gets worse for further increasing numbers of ORs, in particular in the lower utilization scenario with few electives. A possible explanation for this phenomenon could be that a large number of ORs with few electives makes an approximation to the ideal plan of Figure 4 more difficult and thus deteriorates performance. Nevertheless, with the exception of the instances with the lowest number of ORs, average waiting times for emergencies are within 15 minutes for the *breaks* heuristic, which in practice should typically be sufficient for facilitating an immediate service, given that the hospital is informed ahead of time of the emergency. In order to investigate this further, Table 5 shows the percentage of emergencies that can be served within 15 and 30 minutes, respectively, for all procedures.

As can be seen, *breaks* leads to a significantly higher percentage of served emergencies within 15 minutes even when compared to *BIM + ex* and the improvement only slightly decreases with higher utilization. The higher the utilization rate, the lower the percentages of serviced emergencies for all procedures, yet *breaks* still manages to guarantee a ser-

**Table 5** Ratio of emergencies served within 15 and 30 minutes

Util	0.8		0.9		1.0	
	< 15	< 30	< 15	< 30	< 15	< 30
<i>exclusive</i>	0.49	0.55	0.49	0.55	0.49	0.55
<i>BIM</i>	0.75	0.86	0.69	0.81	0.50	0.62
<i>BIM + ex</i>	0.84	0.92	0.81	0.89	0.78	0.87
<i>breaks</i>	0.91	0.94	0.87	0.92	0.83	0.89

**Table 6** Results for stochastic elective surgeries with  $\sigma^* = 10$

Util	#OR	#elec	<i>breaks</i>			<i>BIM</i>			<i>BIM + ex</i>		
			Wait	Over	Overtime	Wait	Over	Overtime	Wait	Over	Overtime
0.8	2	4	20.32	0.00	0.43	27.17	0.00	0.28	18.47	0.00	0.34
0.8	2	6	15.10	0.00	0.46	19.73	0.00	0.32	15.49	0.00	0.32
0.8	2	8	12.07	0.00	0.50	15.37	0.00	0.33	13.25	0.00	0.31
0.8	3	8	10.02	0.43	0.62	18.42	0.00	0.36	10.80	0.00	0.44
0.8	3	12	4.84	0.21	0.67	12.78	0.00	0.35	7.55	0.00	0.39
0.8	3	16	3.22	0.53	0.73	8.32	0.00	0.33	5.86	0.00	0.32
0.8	5	16	1.70	0.57	0.76	16.74	0.00	0.44	7.06	0.00	0.48
0.8	5	24	0.55	0.78	0.83	7.82	0.00	0.39	3.86	0.00	0.41
0.8	5	32	0.56	1.06	0.83	5.26	0.00	0.36	2.57	0.00	0.35
0.8	10	36	0.18	0.96	0.83	12.50	0.00	0.51	5.03	0.00	0.53
0.8	10	54	0.07	1.12	0.87	5.01	0.00	0.44	2.01	0.00	0.44
0.8	10	72	0.04	1.07	0.89	2.96	0.00	0.40	1.03	0.00	0.40
0.8	15	56	0.27	0.53	0.86	10.95	0.00	0.53	4.09	0.00	0.55
0.8	15	84	0.03	1.51	0.87	3.91	0.00	0.43	1.31	0.00	0.43
0.8	15	112	0.01	1.08	0.90	1.91	0.00	0.40	0.59	0.00	0.40
0.9	2	4	22.52	0.00	0.40	33.30	0.00	0.24	20.34	0.00	0.38
0.9	2	6	17.87	0.03	0.40	24.74	0.00	0.26	17.87	0.00	0.30
0.9	2	8	12.89	0.15	0.48	18.33	0.00	0.28	15.09	0.00	0.29
0.9	3	8	12.41	0.87	0.60	23.73	0.00	0.35	13.32	0.00	0.43
0.9	3	12	5.95	0.40	0.65	16.60	0.00	0.30	9.67	0.00	0.34
0.9	3	16	3.74	0.91	0.72	10.72	0.00	0.34	7.14	0.00	0.35
0.9	5	16	8.95	0.48	0.78	22.69	0.00	0.43	9.49	0.00	0.48
0.9	5	24	2.93	0.40	0.81	10.60	0.00	0.39	5.32	0.00	0.41
0.9	5	32	1.45	0.41	0.81	6.86	0.00	0.36	3.50	0.00	0.36
0.9	10	36	3.77	1.10	0.73	17.19	0.00	0.51	6.96	0.00	0.52
0.9	10	54	0.59	1.41	0.80	7.02	0.00	0.43	2.96	0.00	0.44
0.9	10	72	0.28	1.05	0.84	4.09	0.00	0.38	1.47	0.00	0.39
0.9	15	56	5.25	0.74	0.82	16.00	0.00	0.53	6.12	0.00	0.54
0.9	15	84	0.57	1.52	0.81	5.77	0.00	0.43	2.03	0.01	0.44
0.9	15	112	0.17	1.17	0.85	2.65	0.00	0.41	0.89	0.00	0.40
1.0	2	4	25.70	1.19	0.33	41.28	0.00	0.20	23.18	2.88	0.38
1.0	2	6	21.17	1.11	0.33	28.34	0.00	0.23	19.65	3.58	0.30
1.0	2	8	14.54	1.34	0.44	21.91	0.00	0.23	17.16	5.99	0.25
1.0	3	8	15.40	0.53	0.54	29.46	0.00	0.34	15.94	3.87	0.44
1.0	3	12	7.17	0.50	0.62	20.33	0.00	0.33	11.15	4.53	0.36

**Table 6** continued

Util	#OR	#elec	<i>breaks</i>			<i>BIM</i>			<i>BIM + ex</i>		
			Wait	Over	Ontime	Wait	Over	Ontime	Wait	Over	Ontime
1.0	3	16	4.87	0.49	0.69	13.64	0.00	0.31	8.64	5.92	0.33
1.0	5	16	11.12	1.23	0.66	29.00	0.04	0.42	12.19	4.82	0.50
1.0	5	24	4.60	1.02	0.71	13.50	0.00	0.40	6.38	6.15	0.40
1.0	5	32	2.67	0.90	0.75	8.57	0.00	0.35	4.52	7.58	0.35
1.0	10	36	9.44	1.63	0.68	23.23	1.04	0.49	9.41	4.91	0.55
1.0	10	54	3.17	1.90	0.67	9.72	0.72	0.42	3.88	7.07	0.45
1.0	10	72	1.76	2.20	0.68	5.51	0.19	0.38	2.10	8.56	0.37
1.0	15	56	9.19	2.58	0.65	20.70	2.20	0.51	7.93	5.61	0.54
1.0	15	84	2.96	2.49	0.62	8.06	1.83	0.44	2.98	7.04	0.45
1.0	15	112	1.29	2.60	0.62	3.74	1.07	0.41	1.28	7.83	0.41
			6.74	0.94	0.68	14.80	0.16	0.38	8.17	1.92	0.41

**Table 7** Results for stochastic elective surgeries with  $\sigma^* = 20$

Util	#OR	#elec	<i>breaks</i>			<i>BIM</i>			<i>BIM + ex</i>		
			Wait	Over	Ontime	Wait	Over	Ontime	Wait	Over	Ontime
0.8	2	4	21.27	1.01	0.42	28.54	0.00	0.25	19.39	0.13	0.32
0.8	2	6	16.18	1.11	0.44	20.47	0.00	0.29	15.87	0.13	0.29
0.8	2	8	13.01	3.13	0.50	17.06	0.00	0.31	14.23	0.66	0.27
0.8	3	8	10.07	3.12	0.61	18.70	0.00	0.35	10.13	0.00	0.40
0.8	3	12	5.55	1.87	0.65	11.79	0.00	0.35	7.35	0.28	0.36
0.8	3	16	4.30	1.94	0.70	9.14	0.00	0.34	6.10	0.88	0.33
0.8	5	16	2.29	3.36	0.73	13.26	0.00	0.45	5.04	0.00	0.48
0.8	5	24	1.47	4.44	0.77	6.79	0.00	0.39	3.22	1.08	0.38
0.8	5	32	1.32	6.12	0.75	4.74	0.02	0.35	2.40	0.83	0.34
0.8	10	36	0.34	4.92	0.76	6.81	0.10	0.52	2.06	0.13	0.53
0.8	10	54	0.25	6.04	0.77	2.68	0.09	0.42	0.87	0.38	0.41
0.8	10	72	0.22	6.84	0.73	1.66	0.19	0.38	0.59	1.40	0.36
0.8	15	56	0.34	5.60	0.75	5.09	0.03	0.52	1.26	0.18	0.52
0.8	15	84	0.08	6.29	0.75	1.31	0.13	0.45	0.34	0.62	0.44
0.8	15	112	0.04	8.51	0.72	0.73	0.39	0.39	0.18	1.29	0.38
0.9	2	4	24.07	2.26	0.36	34.65	0.00	0.20	21.11	2.31	0.33
0.9	2	6	18.41	2.71	0.40	23.67	0.00	0.27	17.76	2.28	0.29
0.9	2	8	14.70	2.47	0.46	19.83	0.00	0.29	16.01	3.08	0.25
0.9	3	8	13.08	2.17	0.55	22.75	0.00	0.35	12.52	1.55	0.40
0.9	3	12	7.13	4.00	0.62	15.95	0.00	0.31	9.20	4.64	0.35
0.9	3	16	5.71	2.09	0.66	10.97	0.00	0.33	7.24	6.09	0.32
0.9	5	16	7.65	3.88	0.67	18.50	0.10	0.41	7.21	1.96	0.46
0.9	5	24	3.40	3.55	0.69	8.61	0.03	0.39	3.81	2.91	0.40
0.9	5	32	2.71	4.96	0.67	6.23	0.29	0.33	3.17	7.80	0.33
0.9	10	36	2.48	6.28	0.65	10.25	1.21	0.49	3.40	2.77	0.49
0.9	10	54	0.89	8.57	0.65	4.12	1.66	0.41	1.24	5.13	0.42
0.9	10	72	0.69	9.32	0.63	2.22	1.32	0.37	0.75	8.19	0.37
0.9	15	56	2.36	5.30	0.66	7.82	1.55	0.51	2.17	2.64	0.52
0.9	15	84	0.48	9.68	0.64	2.22	1.88	0.45	0.52	4.85	0.44
0.9	15	112	0.31	11.54	0.58	1.14	3.25	0.39	0.26	9.61	0.38

Table 7 continued

Util	#OR	#elec	<i>breaks</i>			<i>BIM</i>			<i>BIM + ex</i>		
			Wait	Over	Overtime	Wait	Over	Overtime	Wait	Over	Overtime
1.0	2	4	26.26	2.58	0.33	39.97	0.00	0.19	22.09	8.32	0.36
1.0	2	6	21.85	3.89	0.33	28.65	0.00	0.23	18.61	16.40	0.30
1.0	2	8	16.91	5.13	0.43	23.96	0.00	0.26	16.93	20.08	0.24
1.0	3	8	17.00	4.36	0.48	30.82	0.00	0.28	14.58	16.39	0.39
1.0	3	12	8.66	4.18	0.58	17.81	0.00	0.34	9.81	14.45	0.39
1.0	3	16	6.68	3.91	0.64	12.76	0.00	0.34	7.72	20.52	0.36
1.0	5	16	9.92	8.37	0.55	23.25	1.92	0.39	8.77	17.10	0.45
1.0	5	24	5.64	7.27	0.59	11.01	0.43	0.37	4.88	22.37	0.40
1.0	5	32	4.29	7.95	0.57	7.24	1.32	0.35	3.64	27.36	0.34
1.0	10	36	6.04	15.06	0.57	14.26	10.94	0.48	4.70	21.12	0.50
1.0	10	54	2.71	15.72	0.53	5.35	9.21	0.42	1.79	25.50	0.42
1.0	10	72	1.97	17.67	0.49	3.10	8.49	0.36	0.94	32.50	0.37
1.0	15	56	4.68	16.45	0.58	11.89	13.25	0.50	3.53	20.61	0.51
1.0	15	84	1.54	19.89	0.50	3.41	15.46	0.44	0.83	28.60	0.44
1.0	15	112	0.99	20.90	0.46	1.55	15.36	0.38	0.39	31.44	0.38
			7.02	6.59	0.59	12.73	1.97	0.37	6.99	8.81	0.39

vice within 30 minutes for about nine out of ten emergencies and more than four out of five emergencies can be serviced within 15 minutes. This is a relevant difference since a service within 15 minutes will typically be sufficient for accepting an emergency.

In order to test the sensitivity of the experimental results against the background of a dynamic and uncertain environment, we also studied the performance of the approaches under the assumption of uncertain surgery times of elective surgeries. For this purpose we introduced a normally distributed error term with a mean of 0 and a standard deviation of 10 and 20 minutes, respectively. Schedules were planned under the assumption of deterministic surgery times, which is typical for practical settings, but the actual surgery lengths of electives were then adjusted by the error term during the simulation, which means that some elective surgeries might have to start later than planned if the OR is not free on time. Tables 6 and 7 display average waiting times for these experiments for *breaks*, *BIM* and *BIM + ex*, since the *exclusive* policy is not affected by changes in elective surgery times. Additionally, the tables report the average unplanned overtime of each OR in minutes (overtime Over), which results from elective surgeries exceeding the limit of the time horizon of 480 minutes, and the percentage of electives that were actually executed on time (Overtime).

The performance measured in average waiting times tends to improve for the *BIM* and *BIM + ex* heuristics with a higher standard deviation of elective surgeries. This is explained by the increased overtime whenever uncharacteristically long surgeries are carried out towards the end of the planning horizon, such that this effectively reduces capacity

utilization within the planning horizon. The performance of *breaks* deteriorates slightly with a higher variance, because unplanned extensions in the duration of elective surgeries strongly interact with the even distribution of breaks over time which overcompensates for the lower capacity utilization, especially in the high variance case.

One can further see that the more even distribution of electives over all ORs and time in *breaks* has the additional benefit of facilitating punctual starts of elective surgeries in the majority of cases. Over all tested medium variance scenarios, 68% of electives start on time, compared to only around 40% for the other procedures. Even in the high variance case, 59% of electives start on time. As a consequence, elective schedules determined by *breaks* will tend to be much more robust not only with respect to the start times of emergencies, but also with respect to unplanned interferences with the start times of elective surgeries.

## 7 Conclusion

In this work, we investigate deterministic scheduling policies that allow a timely service of uncertain emergencies. We showed that evenly distributing non-utilized capacity in the form of breaks over the planning horizon and all resources can significantly reduce waiting times for emergency jobs. Our computational results show that a simple construction heuristic can significantly reduce waiting times in comparison to approaches from the literature. An interesting aspect was furthermore that expected waiting times for emergency jobs are strongly dependent on the combination of number

of resources, number of jobs per resource, and utilization. This might lead to strategies for job mix and scheduling which lead at least for special time intervals or a subset of resources to a good schedule from the point of view of high-priority emergency jobs (this might be useful especially in OR scheduling). Moreover, improvement heuristics such as metaheuristics might improve the results.

In the context of OR scheduling, we have not considered the pre- and post-operative area. However, our policy schedules surgery starts as evenly as possible over the surgery day. If we assume preparation times to be independent of kind and duration of the surgery, then the preoperative workload is smoothed simultaneously. Although surgeries have different durations, we want to have breaks with the same length. So we also smooth the time between two surgery ends and therefore indirectly post-operative area’s workload (compare also Calegari et al., 2020). Dexter et al. (2005) and Marcon and Dexter (2006) considered surgery scheduling’s influence on the post-operative area, especially the PACU. Furthermore, we assumed the interarrival time of emergency patients as constant. In real applications, the interarrival time might be longer in the night hours than in the daytime. This yields to a further question for future research. The transitions from a higher amount of reserved time to a lower and vice versa might be investigated theoretically or in simulation studies.

Hans et al. (2008) scheduled electives with random durations to surgery days. Furthermore, we have not considered how much time should be reserved for emergencies. Approaches according to this are based on historic data (see Lamiri et al., 2008 and van Houdenhoven et al., 2007) or queueing theory (see Zonderland et al., 2010). Both are interesting aspects to include into our setting in future research.

**Funding** Open Access funding enabled and organized by Projekt DEAL.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

### A Proof of Theorem 1

Set  $\rho = \frac{\lambda}{c\mu}$ . The expected number of emergency jobs in the system ( $M/M/c$ ; Barbeau and Kranakis Barbeau and

Kranakis (2007)) is

$$\frac{\rho}{1 - \rho} \cdot \frac{\frac{(c\rho)^c}{c!(1-\rho)}}{\sum_{k=0}^{c-1} \frac{(c\rho)^k}{k!} + \frac{(c\rho)^c}{c!(1-\rho)}} + c\rho,$$

where  $c \in \mathbb{N}$  is the number of resources which are reserved for emergency jobs. With Little’s law (Little, 1961), the expected dwell time in the system is

$$\frac{\frac{\rho}{1-\rho} \cdot \frac{\frac{(c\rho)^c}{c!(1-\rho)}}{\sum_{k=0}^{c-1} \frac{(c\rho)^k}{k!} + \frac{(c\rho)^c}{c!(1-\rho)}} + c\rho}{\lambda}$$

and the expected waiting time

$$\begin{aligned} & \frac{\frac{\rho}{1-\rho} \cdot \frac{\frac{(c\rho)^c}{c!(1-\rho)}}{\sum_{k=0}^{c-1} \frac{(c\rho)^k}{k!} + \frac{(c\rho)^c}{c!(1-\rho)}} + c\rho}{\lambda} - \frac{1}{\mu} \\ &= \frac{1}{c\mu - \lambda} \cdot \frac{\frac{\left(\frac{\lambda}{\mu}\right)^c}{c!(1-\rho)}}{\sum_{k=0}^{c-1} \frac{\left(\frac{\lambda}{\mu}\right)^k}{k!} + \frac{\left(\frac{\lambda}{\mu}\right)^c}{c!(1-\rho)}} + \underbrace{\frac{\frac{\lambda}{\mu}}{\lambda} - \frac{1}{\mu}}_{=0} \\ &= \frac{\frac{\left(\frac{\lambda}{\mu}\right)^c}{c!(1-\rho)}}{(c\mu - \lambda) \left( \sum_{k=0}^{c-1} \frac{\left(\frac{\lambda}{\mu}\right)^k}{k!} + \frac{\left(\frac{\lambda}{\mu}\right)^c}{c!(1-\rho)} \right)} \\ &= \frac{\left(\frac{\lambda}{\mu}\right)^c}{(1 - \rho)(c\mu - \lambda)c! \left( \sum_{k=0}^{c-1} \frac{\left(\frac{\lambda}{\mu}\right)^k}{k!} + \frac{\left(\frac{\lambda}{\mu}\right)^c}{c!(1-\rho)} \right)} \end{aligned}$$

follows. It holds that

$$\begin{aligned} & \frac{\left(\frac{\lambda}{\mu}\right)^c}{c!} \left( \left(1 - \frac{\lambda}{(c+1)\mu}\right) ((c+1)\mu - \lambda) - (c\mu - \lambda) \right) \\ &= \frac{\left(\frac{\lambda}{\mu}\right)^c}{c!} \left( ((c+1)\mu - \lambda) - \frac{\lambda}{(c+1)\mu} ((c+1)\mu - \lambda) - (c\mu - \lambda) \right) \\ &= \frac{\left(\frac{\lambda}{\mu}\right)^c}{c!} \left( \mu - \lambda + \frac{\lambda^2}{(c+1)\mu} \right) > 0 \end{aligned}$$

because  $\mu > \lambda$ . From this

$$\begin{aligned} & \frac{\left(\frac{\lambda}{\mu}\right)^c}{c!} \left( \left(1 - \frac{\lambda}{(c+1)\mu}\right) ((c+1)\mu - \lambda) - (c\mu - \lambda) \right) \\ &+ ((c+1)\mu - \lambda) \frac{\left(\frac{\lambda}{\mu}\right)^{(c+1)}}{(c+1)!} > 0 \end{aligned}$$

and

$$\begin{aligned} & \left(1 - \frac{\lambda}{(c+1)\mu}\right) ((c+1)\mu - \lambda) \frac{\left(\frac{\lambda}{\mu}\right)^c}{c!} \\ & + \left(1 - \frac{\lambda}{(c+1)\mu}\right) ((c+1)\mu - \lambda) \frac{\left(\frac{\lambda}{\mu}\right)^{(c+1)}}{(c+1)! \left(1 - \frac{\lambda}{(c+1)\mu}\right)} \\ & - (1 - \rho)(c\mu - \lambda) \frac{\left(\frac{\lambda}{\mu}\right)^c}{c!(1 - \rho)} > 0 \end{aligned}$$

and

$$\begin{aligned} & \underbrace{\sum_{k=0}^{c-1} \frac{\left(\frac{\lambda}{\mu}\right)^k}{k!} \left( \left(1 - \frac{\lambda}{(c+1)\mu}\right) ((c+1)\mu - \lambda) - (1 - \rho)(c\mu - \lambda) \right)}_{> 0} \\ & + \left(1 - \frac{\lambda}{(c+1)\mu}\right) ((c+1)\mu - \lambda) \frac{\left(\frac{\lambda}{\mu}\right)^c}{c!} \\ & + \left(1 - \frac{\lambda}{(c+1)\mu}\right) ((c+1)\mu - \lambda) \frac{\left(\frac{\lambda}{\mu}\right)^{(c+1)}}{(c+1)! \left(1 - \frac{\lambda}{(c+1)\mu}\right)} \\ & - (1 - \rho)(c\mu - \lambda) \frac{\left(\frac{\lambda}{\mu}\right)^c}{c!(1 - \rho)} \\ & = \left(1 - \frac{\lambda}{(c+1)\mu}\right) ((c+1)\mu - \lambda) \end{aligned}$$

$$\begin{aligned} & \times \left( \sum_{k=0}^c \frac{\left(\frac{\lambda}{\mu}\right)^k}{k!} + \frac{\left(\frac{\lambda}{\mu}\right)^{c+1}}{(c+1)! \left(1 - \frac{\lambda}{(c+1)\mu}\right)} \right) \\ & - (1 - \rho)(c\mu - \lambda) \left( \sum_{k=0}^{c-1} \frac{\left(\frac{\lambda}{\mu}\right)^k}{k!} + \frac{\left(\frac{\lambda}{\mu}\right)^c}{c!(1 - \rho)} \right) > 0 \end{aligned}$$

follow. Further, we get

$$\begin{aligned} & \frac{\left(\frac{\lambda}{\mu}\right)^{(c+1)}}{\left(1 - \frac{\lambda}{(c+1)\mu}\right) ((c+1)\mu - \lambda)(c+1)! \left( \sum_{k=0}^c \frac{\left(\frac{\lambda}{\mu}\right)^k}{k!} + \frac{\left(\frac{\lambda}{\mu}\right)^{(c+1)}}{(c+1)! \left(1 - \frac{\lambda}{(c+1)\mu}\right)} \right)} \\ & < \frac{1}{c+1} \frac{\left(\frac{\lambda}{\mu}\right)^c}{(1 - \rho)(c\mu - \lambda)c! \left( \sum_{k=0}^{c-1} \frac{\left(\frac{\lambda}{\mu}\right)^k}{k!} + \frac{\left(\frac{\lambda}{\mu}\right)^c}{c!(1 - \rho)} \right)}. \end{aligned} \tag{35}$$

Together with

$$\begin{aligned} & \frac{\left(\frac{\lambda}{\mu}\right)^c}{(1 - \rho)(c\mu - \lambda)c! \left( \sum_{k=0}^{c-1} \frac{\left(\frac{\lambda}{\mu}\right)^k}{k!} + \frac{\left(\frac{\lambda}{\mu}\right)^c}{c!(1 - \rho)} \right)} > 0 \end{aligned}$$

the factor  $\frac{1}{c+1}$  in (35) implies the second inequality in

$$\begin{aligned} & 0 \stackrel{(35)}{>} \frac{\left(\frac{\lambda}{\mu}\right)^{(c+2)}}{\left(1 - \frac{\lambda}{(c+2)\mu}\right) ((c+2)\mu - \lambda)(c+2)! \left( \sum_{k=0}^{c+1} \frac{\left(\frac{\lambda}{\mu}\right)^k}{k!} + \frac{\left(\frac{\lambda}{\mu}\right)^{(c+2)}}{(c+2)! \left(1 - \frac{\lambda}{(c+2)\mu}\right)} \right)} \\ & - \frac{\left(\frac{\lambda}{\mu}\right)^{(c+1)}}{\left(1 - \frac{\lambda}{(c+1)\mu}\right) ((c+1)\mu - \lambda)(c+1)! \left( \sum_{k=0}^c \frac{\left(\frac{\lambda}{\mu}\right)^k}{k!} + \frac{\left(\frac{\lambda}{\mu}\right)^{(c+1)}}{(c+1)! \left(1 - \frac{\lambda}{(c+1)\mu}\right)} \right)} \\ & > \frac{\left(\frac{\lambda}{\mu}\right)^{(c+1)}}{\left(1 - \frac{\lambda}{(c+1)\mu}\right) ((c+1)\mu - \lambda)(c+1)! \left( \sum_{k=0}^c \frac{\left(\frac{\lambda}{\mu}\right)^k}{k!} + \frac{\left(\frac{\lambda}{\mu}\right)^{(c+1)}}{(c+1)! \left(1 - \frac{\lambda}{(c+1)\mu}\right)} \right)} \\ & - \frac{\left(\frac{\lambda}{\mu}\right)^c}{(1 - \rho)(c\mu - \lambda)c! \left( \sum_{k=0}^{c-1} \frac{\left(\frac{\lambda}{\mu}\right)^k}{k!} + \frac{\left(\frac{\lambda}{\mu}\right)^c}{c!(1 - \rho)} \right)}. \end{aligned} \tag{36}$$

Let  $p_j \geq 0, j = 0, \dots, \mathcal{O}$ , such that

$$o = \sum_{j=0}^{\mathcal{O}} p_j \cdot j \tag{37}$$

and

$$\sum_{j=0}^{\mathcal{O}} p_j = 1 \tag{38}$$

hold, i.e.  $p_j$  is the probability, or rather the share of time, that  $j$  resources are reserved for emergencies (such a sequence exists, since  $o$  is the mean number of resources reserved for emergencies). If the probabilities  $p_j$  are positive for any number of resources less than  $\lfloor o \rfloor$  or higher than  $\lceil o \rceil$ , the expected waiting time can be reduced by the following procedure:

- (a) Set  $j = \arg \min_{k:\lfloor o \rfloor \neq k \neq \lceil o \rceil \wedge p_k > 0} p_k$ . If  $j < o$ , go to (b), else to (c).
- (b) For any  $k$  with  $p_k > 0$ , and  $k > o$ , set  $p_{j+1} := p_{j+1} + p_j, p_k := p_k - p_j, p_{k-1} := p_{k-1} + p_j$  (such a  $k$  exists due to (37) and the choice of  $j$ ), and  $p_j := 0$ . Then, (37) and (38) are still fulfilled and because of (36) this reduces the expected waiting time. Go back to (a) if any  $\lfloor o \rfloor \neq k \neq \lceil o \rceil$  with  $p_k > 0$  exists.
- (c) For any  $k$  with  $p_k > 0$ , and  $k < o$ , set  $p_{j-1} := p_{j-1} + p_j, p_k := p_k - p_j, p_{k+1} := p_{k+1} + p_j$  (such a  $k$  exists due to (37) and the choice of  $j$ ), and  $p_j := 0$ . Then, (37) and (38) are still fulfilled and because of (36) this reduces the expected waiting time. Go back to (a) if any  $\lfloor o \rfloor \neq k \neq \lceil o \rceil$  with  $p_k > 0$  exists.

Hence, the expected waiting time of emergencies is minimal if always  $o$  resources are reserved for them or if  $o \notin \mathbb{N}_0$ , always  $\lfloor o \rfloor$  or  $\lceil o \rceil$  resources are reserved for emergencies such that the mean number of resources reserved for emergencies is  $o$ .

### B Proof of Theorem 2

We use the mean queue length of an  $M/G/1$  system to compute the expected waiting time and show that it is minimal if the service time’s variance is minimal. Let w.l.o.g. a unit of time be scaled such that  $\lambda = 1$ . Let  $G$  be the distribution of the beforehand scheduled BII’s, i.e. the time between two successive BIMs, and  $D$  the average length of a BII.

The mean queue length (including service) in an  $M/G/1$  system is (Haigh 2013 p. 208 and Asmussen 2003 p. 237)

$$\rho + \frac{\rho^2 + \lambda^2 \text{var}(s)}{2(1 - \rho)},$$

where  $\text{var}(s)$  is the variance of the service time distribution  $s$  and  $\rho = \lambda/\mu$ . With  $D = 1/\mu$  and  $\rho = \lambda D$  this implies

$$\lambda D + \frac{(\lambda D)^2 + \lambda^2 \text{var}(s)}{2(1 - \lambda D)}$$

as the mean queue length (including service). From this, we get the dwell time (Little, 1961)

$$\frac{\lambda D + \frac{(\lambda D)^2 + \lambda^2 \text{var}(s)}{2(1 - \lambda D)}}{\lambda},$$

and the expected waiting time

$$\begin{aligned} \frac{\lambda D + \frac{(\lambda D)^2 + \lambda^2 \text{var}(s)}{2(1 - \lambda D)}}{\lambda} - D &= D + \frac{\lambda D^2 + \lambda \text{var}(s)}{2(1 - \lambda D)} - D \\ &= \frac{\lambda D^2 + \lambda \text{var}(s)}{2(1 - \lambda D)}. \end{aligned}$$

An arriving emergency job must wait even if the queue is empty until the next BIM, so we have to add  $D/2$  weighted with the probability that the queue is empty when an emergency job arrives to the waiting time. The probability of an empty queue is  $1 - \rho = 1 - \lambda D$  (Willig, 1999). We get

$$\frac{\lambda D^2 + \lambda \text{var}(s)}{2(1 - \lambda D)} + (1 - \lambda D) \frac{D}{2}.$$

$\lambda = 1$  implies the waiting time function

$$\begin{aligned} W(D, \text{var}(s)) &:= \frac{D^2 + \text{var}(s)}{2(1 - D)} + (1 - D) \frac{D}{2} \\ &= \frac{D^3 - D^2 + D + \text{var}(s)}{2(1 - D)}. \end{aligned}$$

Hence,  $\text{var}(s) < \text{var}(s')$  implies

$$W(D, \text{var}(s)) < W(D, \text{var}(s')).$$

Since  $\text{var}(s) \geq 0$ , the theorem follows.

### References

Asmussen, S. (2003). *Applied probability and queues* (2nd ed.). New York: Springer.

Barbeau, M., & Kranakis, E. (2007). *Principles of ad-hoc networking*. Chichester: Wiley.

Beliën, J., Demeulemeester, E., & Cardoen, B. (2009). A decision support system for cyclic master surgery scheduling with multiple objectives. *Journal of Scheduling*, 12, 147–161.

Bovim, T. R., Christiansen, M., Gullhav, A. N., Range, T. M., & Hellemo, L. (2020). Stochastic master surgery scheduling. *European Journal of Operational Research*, 285(2), 695–711.

- Calegari, R., Fogliatto, F. S., Lucini, F. R., Anzanello, M. J., & Schaan, B. D. (2020). Surgery scheduling heuristic considering or downstream and upstream facilities and resources. *BMC Health Services Research*, 20, 1–11.
- Cardoen, B., Demeulemeester, E., & Beliën, J. (2010). Operating room planning and scheduling: A literature review. *European Journal of Operational Research*, 201, 921–932.
- de Bruin, A. M., Bekker, R., van Zanten, L., & Koole, G. M. (2010). Dimensioning hospital wards using the Erlang loss model. *Annals of Operations Research*, 178(1), 23–43.
- Dexter, F., Epstein, R. H., Marcon, E., & de Matta, R. (2005). Strategies to reduce delays in admission into a postanesthesia care unit from operating rooms. *Journal of PeriAnesthesia Nursing*, 20(2), 92–102.
- Ferrand, Y. B., Magazine, M. J., & Rao, U. S. (2014). Partially flexible operating rooms for elective and emergency surgeries. *Decision Sciences*, 45(5), 819–847.
- Haigh, J. (2013). *Probability models* (2nd ed.). London: Springer.
- Hans, E. W., Wullink, G., van Houdenhoven, M., & Kazemier, G. (2008). Robust surgery loading. *European Journal of Operational Research*, 185, 1038–1050.
- Heng, M., & Wright, J. G. (2013). Dedicated operating room for emergency surgery improves access and efficiency. *Canadian Journal of Surgery*, 56(3), 167.
- Jung, K. S., Pinedo, M., Sriskandarajah, C., & Tiwari, V. (2019). Scheduling elective surgeries with emergency patients at shared operating rooms. *Production and Operations Management*, 28(6), 1407–1430.
- Kroer, L. R., Foverskov, K., Vilhelmsen, C., Hansen, A. S., & Larsen, J. (2018). Planning and scheduling operating rooms for elective and emergency surgeries with uncertain duration. *Operations Research for Health Care*, 19, 107–119.
- Lamiri, M., Xie, X., & Zhang, S. (2008). Column generation approach to operating theater planning with elective and emergency patients. *IIE Transactions*, 40, 838–852.
- Latorre-Núñez, G., Lüer-Villagra, A., Marianov, V., Obreque, C., Ramis, F., & Neriz, L. (2016). Scheduling operating rooms with consideration of all resources, post anesthesia beds and emergency surgeries. *Computers & Industrial Engineering*, 97, 248–257.
- Little, J. D. C. (1961). A proof for the queuing formula:  $l = \lambda w$ . *Operations Research*, 9, 383–387.
- Marcon, E., & Dexter, F. (2006). Impact of surgical sequencing on post anesthesia care unit staffing. *Health Care Management Science*, 9, 87–98.
- Pham, D.-N., & Klinkert, A. (2008). Surgical case scheduling as a generalized job shop scheduling problem. *European Journal of Operational Research*, 185, 1011–1025.
- Samudra, M., Van Riet, C., Demeulemeester, E., Cardoen, B., Vansteenkiste, N., & Rademakers, F. E. (2016). Scheduling operating rooms: Achievements, challenges and pitfalls. *Journal of Scheduling*, 19(5), 493–525.
- Strum, D. P., May, J. H., & Vargas, L. G. (2000). Modeling the uncertainty of surgical procedure times: Comparison of log-normal and normal models. *Anesthesiology*, 92, 1160–1167.
- van der Lans, M., Hans, E. W., Hurink, J. L., Wullink, G., van Houdenhoven, M., & Kazemier, G. (2006). Anticipating urgent surgery in operating room departments.
- van Essen, J., Hans, E., Hurink, J., & Oversberg, A. (2012). Minimizing the waiting time for emergency surgery. *Operations Research for Health Care*, 1, 34–44.
- van Houdenhoven, M., Hans, E. W., Klein, J., Wullink, G., & Kazemier, G. (2007). A norm utilisation for scarce hospital resources: Evidence from operating rooms in a Dutch university hospital. *Journal of Medical Systems*, 31(4), 231–236.
- van Riet, C., & Demeulemeester, E. (2015). Trade-offs in operating room planning for electives and emergencies: A review. *Operations Research for Health Care*, 7, 52–69.
- van Veen-Berkx, E., Elkhuisen, S. G., Kuijper, B., & Kazemier, G. (2016). Dedicated operating room for emergency surgery generates more utilization, less overtime, and less cancellations. *The American Journal of Surgery*, 211(1), 122–128.
- Vandenberghe, M., De Vuyst, S., Aghezzaf, E.-H., & Bruneel, H. (2019). Surgery sequencing to minimize the expected maximum waiting time of emergent patients. *European Journal of Operational Research*, 275(3), 971–982.
- Vandenberghe, M., Vuyst, S. D., Aghezzaf, E.-H., & Bruneel, H. (2020). Stochastic surgery selection and sequencing under dynamic emergency break-ins. *Journal of the Operational Research Society*, pp. 1–21.
- Vin, J. P., & Ierapetritou, M. G. (2000). A new approach for efficient rescheduling of multiproduct batch plants. *Industrial & Engineering Chemistry Research*, 39(11), 4228–4238.
- Wang, B., Han, X., Zhang, X., & Zhang, S. (2015). Predictive-reactive scheduling for single surgical suite subject to random emergency surgery. *Journal of Combinatorial Optimization*, 30(4), 949–966.
- Wang, H. (2002). A survey of maintenance policies of deteriorating systems. *European Journal of Operational Research*, 139, 469–489.
- Wang, S., Wu, R., Chu, F., & Yu, J. (2020). Identical parallel machine scheduling with assurance of maximum waiting time for an emergency job. *Computers & Operations Research*, 118, 104918.
- Willig, A. (1999). *A short introduction to queueing theory* (p. 21). Telecommunication Networks Group: Technical University Berlin.
- Wullink, G., Houdenhoven, M. V., Hans, E. W., van Oostrum, J. M., van der Lans, M., & Kazemier, G. (2007). Closing emergency operating rooms improves efficiency. *Journal of Medical Systems*, 31(6), 543–546.
- Xiao, Y., & Yoogalingam, R. (2021). Reserved capacity policies for operating room scheduling. *Operations Management Research*, pp. 1–16.
- Zonderland, M. E., Boucherie, R. J., Litvak, N., & Vleggeert-Lankamp, C. L. A. M. (2010). Planning and scheduling of semi-urgent surgeries. *Health Care Management Science*, 13(3), 256–267.

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