J Sched (2018) 21:17–33
https://doi.org/10.1007/s10951-017-0539-8

Case mix classification and a benchmark set for surgery scheduling

Gréanne Leeftink[1] · **Erwin W. Hans**[1]

Published online: 5 September 2017

Abstract Numerous benchmark sets exist for combinatorial optimization problems. However, in healthcare scheduling, only a few benchmark sets are known, mainly focused on nurse rostering. One of the most studied topics in the healthcare scheduling literature is surgery scheduling, for which there is no widely used benchmark set. An effective benchmark set should be diverse, reflect the real world, contain large instances, and be extendable. This paper proposes a benchmark set for surgery scheduling algorithms, which satisfies these four criteria. Surgery scheduling instances are characterized by an underlying case mix, which describes the volume and properties of the surgery types. Given a case mix, unlimited random instances can be generated. A complete surgery scheduling benchmark set should encompass the diversity of prevalent case mixes. We therefore propose a case mix classification scheme, which we use to typify both real-life and theoretical case mixes that span the breadth of possible case mix types. Our full benchmark set contains 20,880 instances, with a small benchmark subset of 146 instances. The instances are generated based on real-life case mixes (11 surgical specialties), as well as theoretical instances. The instances were generated using a novel instance generation procedure, which introduces the concept of "instance proximity" to measure the similarity between two instances, and which uses this concept to generate sets of instances that are as diverse as possible.

✉ Gréanne Leeftink
a.g.leeftink@utwente.nl

✉ Erwin W. Hans
e.w.hans@utwente.nl

[1] Centre for Healthcare Operations Improvement and Research (CHOIR), University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands

Keywords Benchmark set · Classification · Surgery scheduling · Proximity

1 Introduction

The application of Operations Research and Management Sciences to healthcare has been studied since the 1950s and has gained particular attention over the past decade (Hulshof et al. 2012). For elective patients, many key healthcare resources (such as outpatient clinics, diagnostic facilities, and operating rooms) are organized on an appointment basis. Therefore, many studies have looked at the scheduling of appointments or surgical procedures (Brailsford and Vissers 2011; Cardoen et al. 2010; Cayirli and Veral 2003). Brailsford et al. (2009) concluded that the extent of actual implementation of the outcomes of healthcare simulation and modeling studies is disappointingly low, and has always been so. "Startlingly few studies report evidence of implementation, although a relatively large proportion do demonstrate a conceptualized model" (Brailsford et al. 2009). The fact that real-world data is hard to obtain in healthcare may have contributed to Brailsford's conclusion. This unavailability is caused by privacy considerations, and by the simple fact that data registration is primarily done for medical and financial purposes. Although we observe some change, historically no need was felt to record data for operations management purposes. Nevertheless, the application of operations research models and computer simulation inherently requires a lot of data. Therefore, researchers predominantly resort to using theoretical and computer-generated data for their experiments, or use the limited amount of available real-world data, supplemented with computer-generated data. However, the applicability of these outcomes in other settings is questionable. The questions arise: How complex are these instances?

How do these instances compare to instances from other hospitals (and perhaps in other countries)? How does the algorithm perform on such other instances? Finally, how relevant are the presented results for the scientific community, or even for a specific healthcare manager from another hospital?

Benchmark instances are ideal for comparing solution approaches for specific and well-defined problems. A well-known example is the Solomon instances set for benchmarking algorithms for vehicle routing problems (VRP) (Solomon 1987). However, in healthcare scheduling, no widely used benchmark instances exist other than for the nurse scheduling problem (Vanhoucke and Maenhout 2007). Van Riet and Demeulemeester (2014) underline in a recent operating room planning review that test instances are needed that cover realistic hospital settings in order to align the operating room planning research.

Since many healthcare scheduling problems concern assigning patients to shared resources, these problems have a common denominator: scheduling a set of patients with a (stochastic) resource demand. This similarity allows for the creation of benchmark sets, which can be used by a wide variety of algorithms. If specific additional data is needed within a particular application setting, for example additional patient properties such as urgency, users can add such aspects themselves. Having standard benchmark sets available (1) helps to deal with a lack of data availability; (2) allows benchmarking algorithms for similar or identical problem types; and (3) allows for comparing obtained real-life instances to "standard" benchmark instances and for their classification.

One of the most studied topics in healthcare scheduling literature is surgery scheduling, also known as operating room planning and scheduling (Cardoen et al. 2010). However, benchmarking the performance of surgery scheduling between different hospitals is difficult, as case mixes differ between such hospitals. A case mix describes the volume and characterization of all surgery types. Cardoen and Demeulemeester (2011) presented a classification scheme for classifying the surgery scheduling problem. However, the impact of the composition of the case mix on the performance of a surgery scheduling algorithm, which is instance-dependent, was not taken into account.

This paper contributes the following to the literature. We propose a case mix classification scheme, which can also be used to typify and visualize surgery scheduling instances. Furthermore, we illustrate the application of the classification scheme to surgery scheduling datasets obtained from both academic and non-academic hospitals in the Netherlands, and some cases from the literature. We provide a benchmark set for the surgery scheduling problem, which is based on real-life data from Dutch hospitals, as well as theoretical data, and which may serve as a reference benchmark set for testing surgery scheduling algorithms. To ensure that the generated benchmark instances are sufficiently diverse, we introduce the concept of "instance proximity", which is a measure for the similarity of instances. We introduce an instance proximity maximization approach for the instance generation procedure. Finally, we provide applications with which unlimited additional instances and samples can be generated, using statistical distributions based on real-life and theoretical data.

The remainder of this paper is organized as follows. Section 2 discusses the literature about benchmarking, instance generation, and instance classification. In Sect. 3, we describe the problem definition and present the case mix classification. Section 4 applies the classification to real-life datasets and datasets from the literature. Section 5 presents the instance generator, and introduces the instance proximity maximization concept. Finally, in Sect. 6 we present our conclusions.

2 Literature

The classification of scheduling instances, and the development and use of benchmark sets, has been an important topic in the operations research literature (Kolisch et al. 1999). To develop ideas for how to classify patient scheduling instances, we investigate the literature about benchmark sets and instance generation for planning and scheduling problems in Sect. 2.1. In Sect. 2.2 we discuss the conditions for a benchmark set to be effective.

2.1 Benchmark sets

A benchmark set is defined as a collection of instances of a class of combinatorial optimization problems (Kolisch et al. 1995). A benchmark set is also referred to as an instance library. A well-known and widely used benchmark set for the VRP problem was provided by Solomon (1987). Researchers use this benchmark set and adapt the instances to their specific needs, by adding their own characteristics. A widely used extension of the Solomon benchmark set was developed by Gehring and Homberger (2001), who added residual groups to the existing instances. Kok et al. (2010) used the Solomon instances set and added time-dependent travel times and driving regulations to make them feasible for common, practical situations. More recently, Pillac et al. (2013) presented a set of technician scheduling problem instances, extended from the Solomon instances set, by adding technician crews.

Benchmark sets also exist (among others) for project scheduling problems (Kolisch and Sprecher 1996; Wauters et al. 2016), timetabling problems (Qu et al. 2009), and personnel scheduling problems (Musliu et al. 2004).

A project scheduling problem library (PSPLIB) was presented by Kolisch and Sprecher (1996), which among others exists of instances generated by the project scheduling

instance generator ProGen (Drexl et al. 2000; Kolisch et al. 1995). In this library, several benchmark sets are available for researchers, who can use the benchmark set of their needs. These benchmark sets are updated over the years, depending on the progress in the project scheduling research field (Kolisch and Sprecher 1996). A combination of instances from PSPLIB, added with release dates and global resources, were used as instances for the MISTA 2013 challenge (Wauters et al. 2016). Also, a combination of RCPSP instances was used to generate multi-project scheduling instances, which are known as MPSPLIB (Homberger 2012).

In their literature review on timetabling, Qu et al. (2009) analyzed and characterized the available timetabling benchmark sets. They summarized the applied techniques and corresponding results on the benchmark sets. They concluded that benchmark sets should be updated and extended according to the needs of the research area and derived from real-life data, in order to minimize the gap between theory and practice.

Multiple benchmark sets exist for personnel scheduling. The University of Nottingham presents an overview of various personnel scheduling benchmark sets, derived from researchers and from industry (Curtois 2016). Musliu et al. (2004) also provided personnel scheduling instances, which were generated by selecting a solution and randomly generate an instance based on this solution. This way, they already have a (near) optimal solution for each of their generated instances.

Although various benchmark sets exist for well-known combinatorial optimization problems, for healthcare planning and scheduling there only exist various benchmark sets for nurse scheduling (Brucker et al. 2010; Musliu et al. 2004; Vanhoucke and Maenhout 2009) and patient admission scheduling (Bilgin et al. 2012). Typically, these benchmark sets, such as NSPLib (Vanhoucke and Maenhout 2009), are not real-world-based instances but are generated randomly. Within the nurse scheduling research field, two competitions have been organized, for various problem configurations, such as multi-stage nurse rostering (Ceschia et al. 2015). In the first competition, three tracks were presented, based on the available running time of the algorithms, including mall, medium, and large sized instances to solve (Haspeslagh et al. 2014). A patient admission scheduling benchmark set (Bilgin et al. 2012) has been generated on the basis of a single real-life instance from Demeester et al. (2010). Since only limited real-life data were available, Ceschia and Schaerf (2016) presented a benchmark set for patient admission scheduling, together with an instance generator, solution validator, and first solutions. The instances are generated based on randomly generated theoretical case mixes.

To the best of our knowledge, no widely used benchmark sets have been reported for other healthcare scheduling problems, such as the surgery scheduling problem (Riise and Burke 2011). Reasons for this may be the lack of data due to reluctance of hospitals to disclose data, the numerous different representations of essentially the same problem, and the many ways of evaluating solution procedures (Vanhoucke and Maenhout 2007). In the remainder of this paper, we focus on generating and providing benchmark instances for the most studied healthcare problem, namely the surgery scheduling problem. We refer to Denton et al. (2010) for an extensive problem description of the deterministic and stochastic surgery scheduling problem in multiple operating rooms (ORs).

2.2 Conditions for benchmark sets

An algorithm may perform well on one instance, but can have a poor performance on another comparable instance. To provide a benchmark set that represents real-world problems, but also allow for an instance-independent comparison of the performance of algorithms, benchmark sets need to systematically integrate the characteristics of the problem as their parameters (Kolisch et al. 1995). According to Vanhoucke and Maenhout (2007), a benchmark set should satisfy four conditions: diversity, realism, size, and extendibility. First, the instances of a benchmark set should be as *diverse* as possible, to facilitate unbiased evaluation. They should cover the full range of complexity. Second, a benchmark set should reflect *real-world* problems. Burke et al. (2004) stated there is a critical need to use real-world data more frequently for the nurse scheduling problem, to increase the implementation of algorithms in practice. Third, the *size* of the benchmark instances should be large enough to facilitate meaningful statistical analyses. The instance size and solvability trade-off makes a mix of smaller and larger sized instances (and hence easier and harder computational analyses) a promising option. Finally, a benchmark set should be easily *extendable* with other features by other researchers (Vanhoucke and Maenhout 2007).

Where realism, size, and extensibility are independent of the instance characteristics, a diverse benchmark set needs the identification of instance characteristics. Different combinations of instance characteristics, such as the variation of the surgery duration, result in different complexities and different solutions. Therefore, instance characteristics need to be identified, so as to be able to generate and classify diverse instances. This characterization should be generic, and applicable to any instance (Vanhoucke and Maenhout 2009).

Any study to all instance characteristics will be very time-consuming. An alternative is to generate a benchmark set that includes various instance types within a specific area (Kolisch et al. 1999). The literature describes a limited number of instance generators for systematically generating instances, such as (extended versions of) ProGen (Drexl et al. 2000;

Kolisch et al. 1995), DaGen (Agrawal et al. 1996), and RanGen (Demeulemeester et al. 2003). All these examples are of instance generators for project scheduling.

In conclusion, numerous benchmark sets exist for combinatorial optimization problems. However, for healthcare scheduling problems only a few benchmark sets exist, and no benchmark sets have been developed for surgery scheduling problems. The patient admission scheduling set of Ceschia and Schaerf (2016) is the closest to a surgery scheduling benchmark set available. An effective benchmark set should satisfy four conditions: diversity, realism, size, and extensibility.

3 Classification of surgery scheduling instances

Before generating benchmark instances, we first introduce the characteristics of surgery scheduling instances and the underlying case mix (Sect. 3.1) and then introduce a classification method for surgery scheduling instances (Sect. 3.2). Finally, we give some examples of how to incorporate specific surgery scheduling problem settings in the instances, using various instance configurations (Sect. 3.3).

3.1 Case mix and surgery scheduling instance characteristics

In this subsection, we give a formal description of the characteristics of surgery scheduling instances and the case mix they are based on.

We aim to include only generic instance parameters in our benchmark set, to allow other researchers to easily extend the set to include problem-specific parameters. We refer to the surgery scheduling problem classification of Cardoen and Demeulemeester (2011) for an extensive overview of specific surgery characteristics that can be incorporated as additional parameters of scheduling instances.

Underlying a surgery scheduling instance is the hospital's case mix, which describes the volume and properties of all surgery types that the hospital performs. A surgery type has a duration distribution, and is performed by (surgeons from a) surgical specialty. The 3-parameter lognormal distribution is proven to have the best fit with surgery duration distributions (May et al. 2000; Stepaniak et al. 2009). Therefore, for each surgery type $t \in T$, we use three parameters $\mu_t \in (0, \infty)$, $\sigma_t \in (0, \infty)$, and $\gamma_t \in [0, \infty)$ corresponding with the mu, sigma, and threshold (location) of the 3-parameter lognormal distribution. Using these parameters, an average duration m_t and standard deviation s_t can be calculated for each surgery type $t \in T$, using the following formulas:

$$m_t = \gamma_t + e^{\mu_t + \frac{\sigma_t^2}{2}} \tag{1}$$

$$s_t = \sqrt{\left(e^{\sigma_t^2} - 1\right) * e^{2*\mu_t + \sigma_t^2}} \tag{2}$$

In addition to a duration distribution, a surgery type $t \in T$ has a normalized relative frequency $f_t \in [0, 1]$ in the case mix ($\sum_t f_t = 1$). So, if $f_t = 0.01$, then on average 1% of all surgeries is of type t.

With a given case mix, one can generate instances of any desirable size. To generate a surgery scheduling instance requires a number of operating room blocks of given duration (e.g., 15 operating rooms of 8 h), and a load parameter $\alpha \in [0, \infty)$ that determines the total expected surgery workload. For example a load of $\alpha = 0.9$ means that the total expected surgery duration equals 90% of the total operating room block durations. An instance with 15 operating room blocks of 8 h and $\alpha = 0.9$ implies a total expected surgery workload of $0.9 * 15 * 8 = 108$ h. It is now straightforward how to generate an instance with a total expected workload of 108 h and given case mix characteristics. Reversely, given a set of surgeries, we can calculate the total expected surgery workload by adding all expected surgery durations.

In conclusion, a surgery scheduling instance is based on a case mix, which is a collection of surgery types with a volume and duration distribution. A surgery scheduling instance consists of a list of surgeries with a corresponding surgery type. Each surgery type has a given frequency and duration distribution. An instance also contains a number of OR blocks of given equal capacity and target load, to which surgeries should be assigned.

3.2 Classification of surgery scheduling instances

Section 3.1 explained that an instance originates from a case mix. Given a particular case mix, unlimited instances can easily be generated randomly. As each OR department has its own case mix, a surgery scheduling benchmark set is only complete, if it encompasses the diversity of prevalent case mixes. This raises the need to classify case mixes. We propose a classification based on the surgery type duration and the coefficient of variation. Both parameters are indicators of the complexity of a scheduling problem.

The surgery type duration divided by the OR block capacity is an indicator of the scheduling flexibility. Instances where most surgeries have a high duration, lead to schedules with gaps, since there are not enough short duration surgeries to fill these gaps. In our experience, a thorax surgery unit is such an example. Here, surgery durations are typically 4–6 h within 8 h working days. Contrarily, outpatient surgery departments typically have low surgery durations with a high repetition, thus allowing dense OR-schedules to be made.

The coefficient of variation is an indicator of the variability of a system and equals the standard deviation divided by the mean (Tyler et al. 2003). A higher coefficient of variation

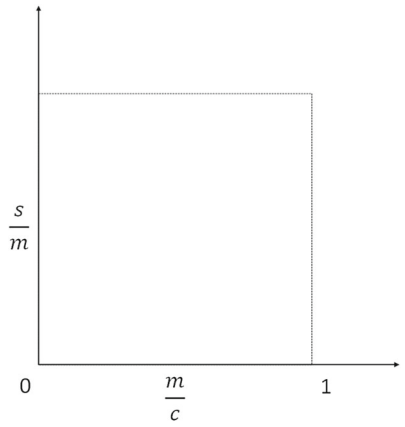

Fig. 1 Case mix classification visualization

indicates high variability in surgery duration, which leads to more uncertainty in the realization of instances. Therefore, it affects the performance of realized schedules, for example in terms of overtime, utilization, or cancellations. This effect may necessitate applying a more robust approach, in which (for example) scheduled slack time is used to alleviate the effects of uncertain surgery durations (Hans et al. 2008). A low coefficient of variation results in easier scheduling problems with almost no risk of incurring overtime and a high probability of fully utilizing OR-blocks, for example in an outpatient OR department (Tyler et al. 2003).

To classify case mixes based on these parameters, we propose the visualization shown in Fig. 1. The x axis is the expected duration (m) of the surgery type in relation to the total capacity of one OR-block (c). The y axis corresponds to the coefficient of variation ($\frac{s}{m}$) in surgery type durations. Note that besides case mixes, instances can also be plotted in the case mix visualization.

In addition to the case mix classification, we define so-called case mix profiles, which are partitions of the case mix classification. We consider 17 case mix profiles, each of which is such a partition. Figure 2 shows 16 of these case mix profiles; case mix profile 0 is the one in which all surgery types are included.

3.3 Surgery scheduling problem settings

Specific surgery scheduling problem settings can be introduced using various instance configurations. We discuss three examples below.

First, patient characteristics, such as urgency, can be taken into account (Cardoen et al. 2010). For example, emergency patients may be classified as urgent patients. They typically have a higher variation in surgery durations. Therefore, the case mix underlying the instances representing emergency departments, could be case mix profile 1 or 2, or the combination of both, case mix profile 7.

Second, both block scheduling and open scheduling systems can be analyzed (Denton et al. 2007). With block scheduling, a range of medical disciplines can be modeled using a combination of instances with varying case mixes. Medical disciplines can be modeled using a specific case mix per discipline. For open scheduling, the combination of several disciplines can be modeled by combining the instances following from each medical discipline. When block scheduling is considered, each medical discipline has its own instance with a set number of ORs, which can be optimized independently of other disciplines. The same approach can be applied to larger planning units, such as hospitals. Instances with different underlying case mixes can be combined to derive a specific combination of surgeries for a given number of ORs.

Finally, researchers can choose the amount of uncertainty incorporation (Cardoen and Demeulemeester 2011). Researchers may choose to only consider the deterministic realizations, or use the 3-parameter lognormal distribution underlying each surgery.

4 Example application of case mix classification

In this section, several case studies from both the literature as well as from practice are identified and classified based on the instance classification proposed in Sect. 3.

Marcon et al. (2003) simulated the operating theater in order to master the risk of no realization and to maximize the operating rooms' utilization time. To analyze the performance of their approach, they generated instances consisting of surgeries with a mean case duration (contained between 60 and 180 min in multiples of 10), and a standard deviation (between 10 and 50% of the case duration). Their operating room opening hours were 8 h per OR, which clearly positions the case mix of this paper in the lower-left quadrant (case mix profile 3), as shown in Fig. 3. This case mix gives the opportunity to derive high performances on different performance indicators, for example in terms of utilization, compared to case mixes with higher coefficients of variations or longer surgery durations (Tyler et al. 2003). Lamiri et al. (2009) developed methods for operating room planning with shared capacity. Their instances were based on surgeries with a uniform distributed duration on the interval [0.5, 3.0]. Since no standard deviation was included, the case mix can only be plotted at the horizontal axis, as shown in Fig. 4. The operating room opening hours were 8 h a day, which positions this case mix in the lower-left quadrant as well.

Many studies exist where authors based test instances on real-life data from partnering hospitals. Marques et al. (2012) developed a model to maximize the utilization of operating room theaters. Their instances were based on real-life data from a hospital in Portugal, as shown in Fig. 5. Hans et al.

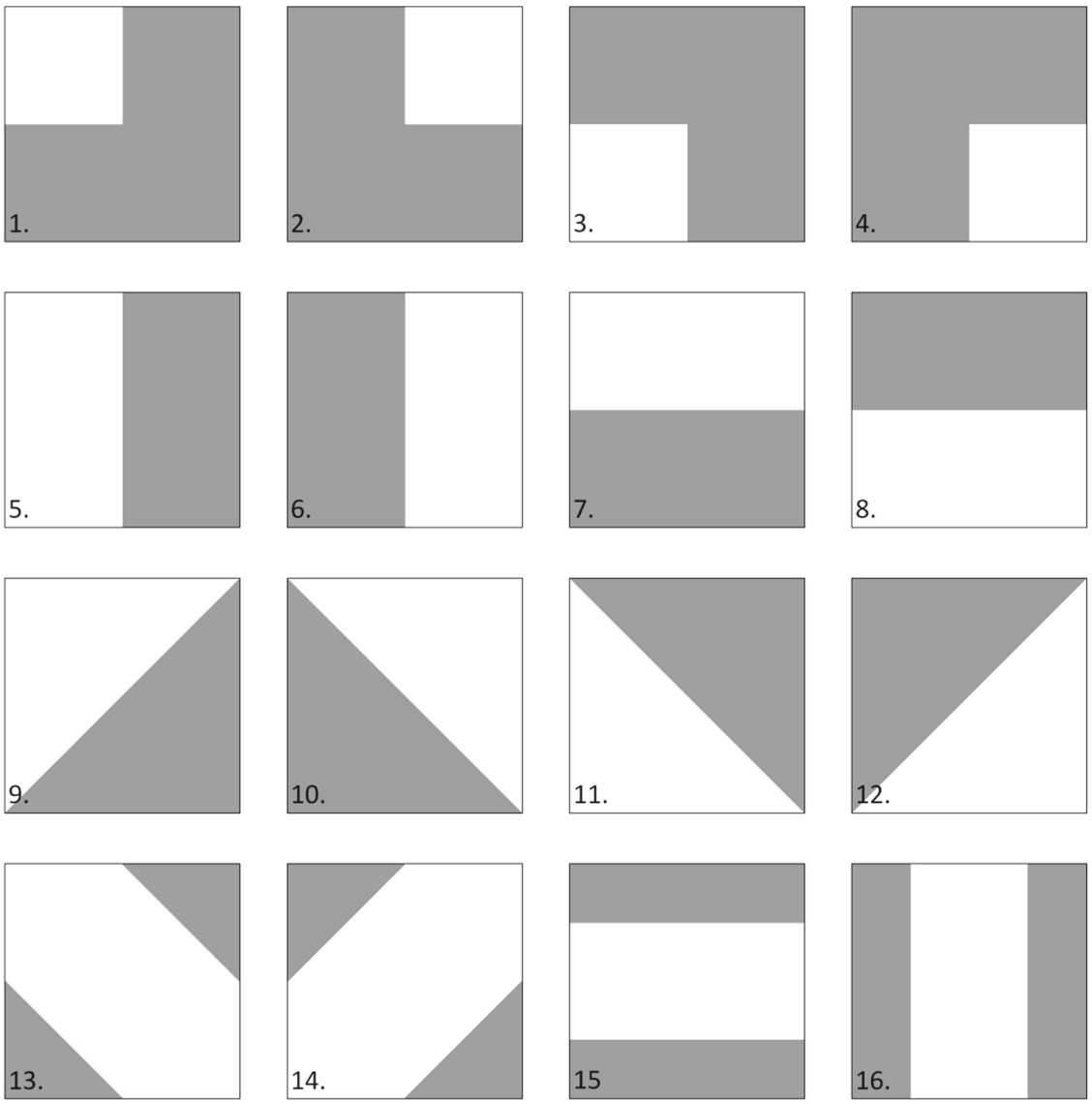

Fig. 2 Theoretical case mix profiles. The *white area* indicates what surgery types are included in the case mix, following the case mix classification in Fig. 1. The X- and Y-axes range between 0 and 1. *Note*: Case mix profile "0" is omitted—it contains all surgery types

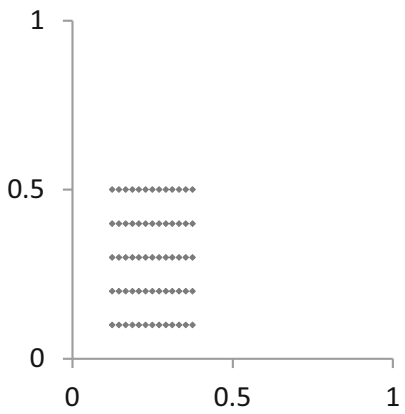

Fig. 3 Case mix plot (Marcon et al. 2003)

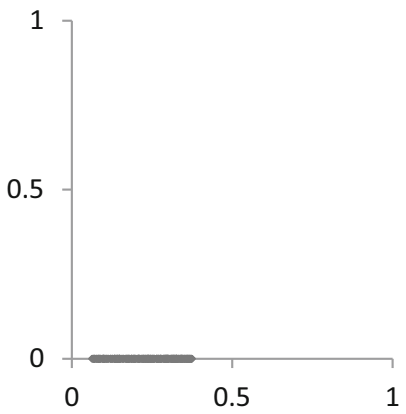

Fig. 4 Case mix plot (Lamiri et al. 2009)

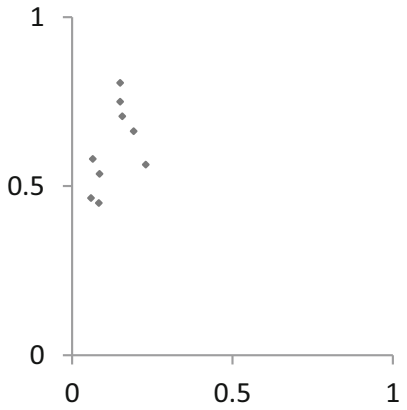

Fig. 5 Case mix plot (Marques et al. 2012)

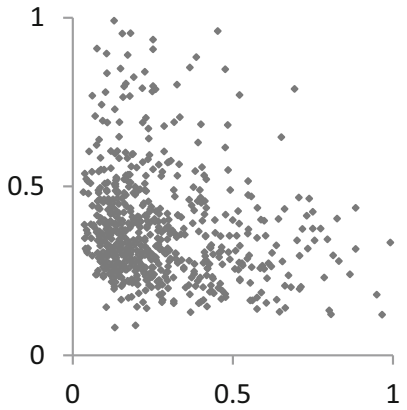

Fig. 6 Case mix plot Erasmus MC overall

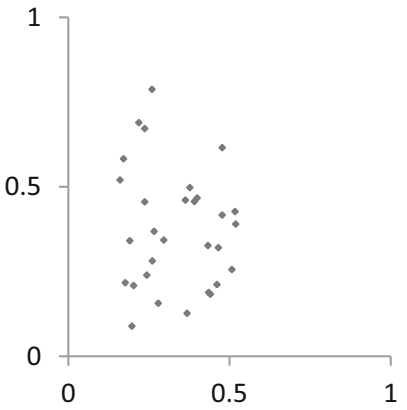

Fig. 7 Case mix plot Erasmus MC specialty 1

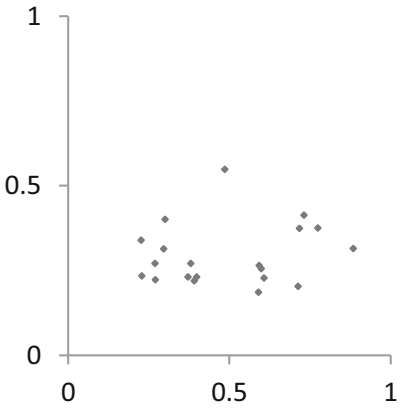

Fig. 8 Case mix plot Erasmus MC specialty 2

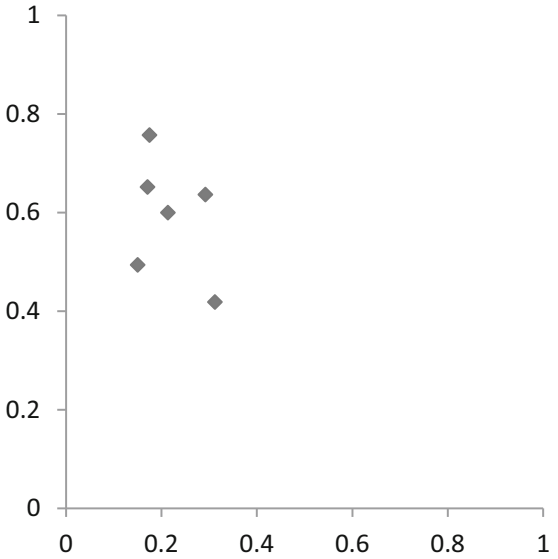

Fig. 9 Case mix plot Norwegian hospital (Mannino et al. 2014)

(2008) published a paper on the robust surgery loading for surgery scheduling using instances based on 10 years of data of all elective inpatient surgeries in Erasmus MC, a large academic hospital in the Netherlands. Using this dataset, not only the hospital's case mix can be determined, but also the case mix per specialty, as shown in Figs. 6, 7, and 8. These figures show that Erasmus MC has many small surgeries compared to their opening hours, but that there are some large differences between specialties. Riise and Burke (2011) used real-life data from a Norwegian Hospital in order to generate test cases for their method for surgery admission planning (Mannino et al. 2014), as shown in Fig. 9. Even though the dataset contained many realizations, the surgery types were not as specific as in the Erasmus MC dataset.

Furthermore, we analyzed some datasets from both academic and non-academic hospitals in the Netherlands. The case mix of a specialized hospital is shown in Fig. 10. This hospital is dedicated to orthopedic surgeries, which corresponds with case mix profile 3. Figure 11 shows the case mix of a general hospital, and the case mix of a university hospital is shown in Fig. 12. As expected by the more complex nature of surgeries performed in an academic hospital, their case mix has a higher coefficient of variation compared to the general hospital case mix.

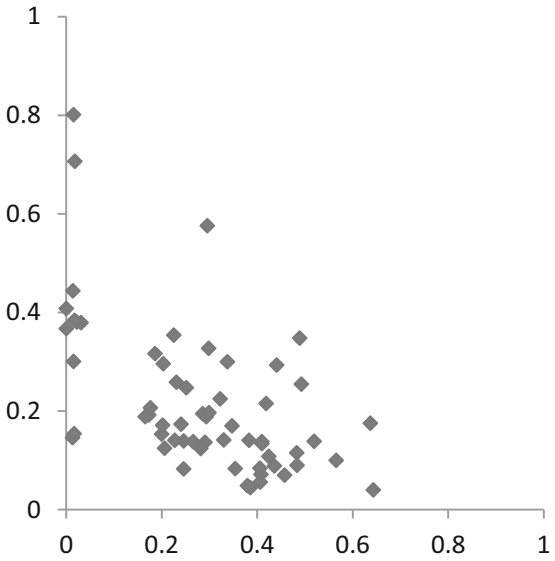

Fig. 10 Case mix plot specialized hospital

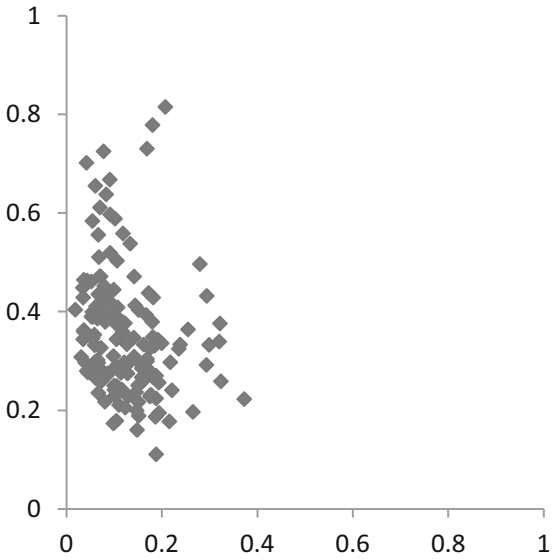

Fig. 11 Case mix plot general hospital

As one can see, the performance of surgery scheduling algorithms for case mixes such as the first specialty of Erasmus MC, or a specialized hospital, can be analyzed using, for example, the generated data of Marcon et al. (2003). However, for an academic hospital, such as the Norwegian hospital, the data of Marques et al. (2012) is more appropriate. Therefore, the case mix of an algorithm's instance should be classified in order to apply the results in generic or real-life settings.

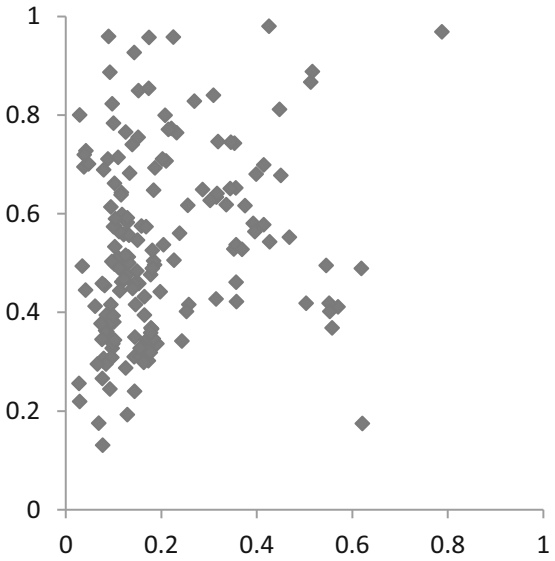

Fig. 12 Case mix plot university hospital

5 Benchmark set for surgery scheduling

A benchmark set should contain a diverse selection of instances (Vanhoucke and Maenhout 2007). In Sect. 3 we saw how diverse instances can be identified based on their underlying case mix. This allows for generating diverse instances. This section presents the parameter settings and describes the generation of benchmark instances.

Based on a case mix and the surgery types' characteristics, we can generate surgery scheduling instances, by drawing a number of corresponding surgeries. Multiple surgery scheduling instances can be combined to form an instance where surgeries of different specialties are planned in shared operating room blocks (e.g., the open block scheduling strategy), or a surgery scheduling instance forms an instance for a surgical specialty that schedules surgeries in its own operating room blocks (e.g., the closed block scheduling strategy).

To provide a benchmark set that is applicable to a broad range of surgery scheduling problems, all blocks generated have equal capacity. When an instance is needed with various block capacities, multiple instances can be combined, as explained in Sect. 3.3. All time-related data, such as surgery durations, are scaled to [0, 1], in which "1" represents the default capacity of one block.

We generate two datasets of underlying surgery types. The first (Sect. 5.1) is based on a collection of real-life data from different hospitals throughout the Netherlands. The second (Sect. 5.2) is based on theoretical (generated) data. We will use these two datasets to generate benchmark sets, which consist of instances, each of which is a set of surgeries. Before presenting the final benchmark sets in Sect. 5.4, in Sect. 5.3, we present our instance generation and selection approach.

Here, we will present a novel technique to generate and select instances in such a way, that the resulting instances in the benchmark set are sufficiently diverse.

5.1 Real-life instance generation

Table 1 shows the experiment design for the instance generation of the real-life benchmark set. We consider the eleven specialties shown in Table 1, which are the most common in practice in our experience. The surgery types underlying the dataset are derived from data from multiple academic and non-academic hospitals throughout the Netherlands over the past 10 years. To consider uniformed surgeries, we chose to include the surgery type ID, the specialty, the frequency of this surgery per year, and the μ, σ, γ, m, and s of each surgery type. Note that the relative frequency per specialty per year is considered. For ease of interpretation, in the benchmark instances, the unit of the surgery type duration distributions is 'minutes'. Without loss of generality, we set the OR capacity (c) to the common value of 480 min.

To determine the surgery types based on historical data of performed surgeries, we need to cluster individual surgeries into logistically similar surgery types. The lowest level clusters are clusters based on surgical procedure codes that correspond to patient types. As coding systems differ between countries, a higher-level clustering based on logistical characteristics can be applied to allow the comparison of case mixes (e.g., surgeon or surgical specialty, surgery duration).

We have derived over 1000 surgery types from almost 200,000 surgery realizations from 5 hospitals, from recent years. For each anonymized surgery in this set, the surgical specialty, treatment code, and duration realization was provided. We cluster these surgery realizations based on their specialty and treatment code, a commonly used classification in the Netherlands to indicate planning characteristics. To obtain surgery types, we fit a 3-parameter lognormal distribution to each cluster of surgery realizations, using a mean square error (MSE) minimization procedure. Surgery types were only included when more than 20 realizations are available, and when the derived distribution's MSE is smaller than 0.001. Table 2 shows the statistics of the outcome.

We consider eight values for the number of ORs ($j \in \{5, 10, 15, 20, 25, 30, 35, 40\}$). Surgery scheduling in practice typically has a planning horizon anywhere between a single day and two weeks. In almost all hospitals we collaborate with, the planning horizon is one week, during which specialties typically have up to 20 blocks of surgery time.

We consider ten values for the ratio of surgery load to ORs ($\alpha \in \{0.80, 0.85, \ldots, 1.20\}$). An instance is considered of a certain surgery load if the deviation of the surgery load is less than 0.025. For example, an instance with surgery load $\alpha = 0.80$, should have a workload between 0.775 and 0.825.

For each specialty, we provide a total of X instances. The instances are built by repeatedly selecting a random surgery type, for which we add one patient (surgery) to a set of patients until the ratio of surgery load to ORs deviates less than 0.025 from the desired α. While the ratio is less than α, we do 100 attempts to add another random patient, who is added iff the resulting ratio is closer to the desired α. Accordingly, we generate $3 * X$ instances, from which we finally select X instances for the benchmark set. The instance generation procedure is summarized in Box 1. In Sect. 5.3 we explain how we perform this selection, aiming to select the X most diverse instances.

With $X = 10$ instances for all combinations of parameters, the final real-life benchmark set consists of 8800 instances.

5.2 Theoretical instance generation

In this section we describe how we generated the benchmark sets with theoretical instances. Instead of generating instances for specialties (for the real-life instances), here we focus on the case mix profiles described in Sect. 3.2.

Table 3 shows the experiment design for the generation of theoretical instances, which leads to 1360 instance parameter combinations. In comparison with the real-life instances, observe that the only difference is that here we use 17 case mix profiles, instead of 11 specialty case mixes. The instance generation procedure is almost the same as in the previous section. The difference lies in how we select a random surgery type. For the theoretical benchmark instances, we generate a random surgery type for every surgery we (try to) add to an instance. The procedure is as follows. For each surgery t we sample a random coordinate (X_t, Y_t) from the case mix profile at hand. Consequently, X_t is the expected duration in relation to the OR capacity ($\frac{m_t}{c}$), and Y_t is the coefficient of variation ($\frac{s_t}{m_t}$). So, we have that $m_t = c \cdot X_t$ and $s_t = X_t \cdot Y_t$. Now, from m_t and s_t we must determine the three parameters $\mu_t \in (0, \infty), \sigma_t \in (0, \infty)$, and $\gamma_t \in [0, \infty)$ corresponding with the mu, sigma, and threshold (location) of the 3-parameter lognormal distribution. Equations (1) and (2) describe the relation between m and s and μ_t, σ_t, and γ_t. Since we have one unknown parameter too many to solve the equations, we choose to randomly set $\gamma_t = 0.75 \cdot R \cdot m$, where R is a random number from [0, 1]. This formula follows from our analysis of the threshold parameters of the real-life surgery types used in the previous section, which we found to lie between 0 and 0.75.

The instance generation is equal to the procedure of the real-life instance generation, as summarized in Box 1. Again, $3 * X$ instances per combination of parameters are generated, from which the X most diverse instances are selected, as explained in Sect. 5.3. With $X = 10$ instances per parameter combination, the theoretical benchmark set con-

Table 1 Real-life instance generation design (Demirkol et al. 1998)

Experimental design for real-life instance generation		
Problem parameter	Values considered	Number of values
Specialty (p)	CHI, ORT, ENT, GYN, PLA, URO, EYE, THO, ONC, NEU, MIX	$\|P\| = 11$
Number of ORs (j)	5, 10, 15, 20, 25, 30, 35, 40	$\|J\| = 8$
Load (α)	0.80, 0.85, . . ., 1.20	$\|\mathrm{A}\| = 10$

Total instance parameters: 880

Table 2 Surgery realizations underlying real-life specialties

Specialty short name	Specialty full name	# Surgery types	# Surgery realizations
CHI	General surgery	149	9311
ENT	Otolaryngology	146	11,986
EYE	Ophthalmic surgery	91	7953
GYN	Obstetric and gynecologic surgery	60	4116
MIX	Remaining specialties, such as colorectal surgery, pediatric surgery, trauma surgery, vascular surgery, etc.	173	46,938
NEU	Neurological surgery	47	2832
ONC	Surgical oncology	43	6466
ORT	Orthopedic surgery	133	7618
PLA	Plastic surgery	73	3022
THO	Thoracic surgery	28	2248
URO	Urology	75	5627
Total		1018	108,117

Table 3 Theoretical instance generation design (Demirkol et al. 1998)

Experimental design for fictitious instance generation		
Problem parameter	Values considered	Number of values
Case mix profile (p)	0, 1, . . ., 16	$\|P\| = 17$
Number of ORs (j)	5, 10, 15, 20, 25, 30, 35, 40	$\|J\| = 8$
Load (α)	0.80, 0.85, . . ., 1.20	$\|\mathrm{A}\| = 10$

Total instance parameters: 1360

sists of 13,600 instances. In total, the real-life and theoretical instances in the benchmark set amount to 22,400 instances.

5.3 Instance proximity maximization

Our aim is to generate a benchmark set with instances that are mutually significantly different. A benchmark set serves to give insight into the problem characteristics that make it hard for an algorithm. An instance that is very similar to another instance in the set thus provides no additional insights. To the best of our knowledge, there exists no measure for assessing the similarity of instances, besides data mining techniques. We therefore propose the following approach to measure similarity of instances, which we shall refer to as *instance proximity*.

Consider two instances A and B, which were generated based on the same case mix, which have the same load and the same number of ORs. As an instance consists of a set of surgeries, we need to assess the similarity of surgeries in instance A and B. Surgeries are characterized by 5 aspects: the expected duration, duration standard deviation, three parameters of the lognormal duration distribution (see Sect. 3.1). To assess the similarity of two surgeries—say S1 and S2 from respectively instance A and B—we could either take a deterministic or stochastic approach. A deterministic approach would only consider the expected duration. One could for example say that if the absolute difference between the expected durations of S1 and S2 is below a threshold, the surgeries are regarded proximal. In a stochastic approach, we would consider the duration distribution characteristics of S1 and S2. In this case, two surgeries are regarded proximal

Box 1 Instance generation procedure

Given:

- *a case mix,*
- *number of ORs,*
- *a set of loads* ($\alpha\varepsilon\{0.80, 0.85, \ldots, 1.20\}$)

The procedure will generate 3X instances for each of the loads in the set.

Step 1: Select a random surgery type.

Step 2: Add a surgery of this type to the instance, and determine the load (total expected surgery duration / total OR capacity) of the instance.

Step 3:

If the load deviates less than 0.025 from any load α in the set, continue with Step 4.

Else

if the load is more than the highest load $\alpha + 0.025$, discard the instance and continue with Step 6.

Else repeat Step 1-3.

Step 4: If the load is less than the desired load α, do 100 attempts to add another random surgery, which is added iff the resulting load is closer to α.

Step 5: Save the resulting instance for the desired α. If a load α. has sufficient instances, remove it from the set of loads.

Step 6: Repeat the generation procedure until for each load α. in the set, $3X$ instances are generated.

if the overlap of the duration distributions' density functions is above a threshold. Observe that the deterministic approach discriminates less than the stochastic approach. For example, two surgeries may be considered proximal from a deterministic point of view, while not being considered proximal from a stochastic point of view (i.e., their expected durations are proximal, but distribution functions overlap insufficiently). Since using a deterministic approach will yield less proximate instances, in the remainder we shall apply the deterministic approach. We introduce the following definition:

Definition 1 Surgeries S1 and S2 are ε-proximate iff their expected durations differ less than $\varepsilon\%$.

In order to select those instances that are maximally different, we first analyze the ε-proximity of all surgery pairs between two instances. Observe that surgeries can be ε-proximate to more than one other surgery from the other instance. Therefore, to determine how proximal two instances are, we have to find the maximum matching of ε-proximate surgeries, which can be found in polynomial time. The so-called ε-proximity quantity of two instances is determined as follows:

Definition 2 The ε-proximity quantity of two instances equals the total workload of all ε-proximate surgeries selected in the maximum matching from both instances, divided by the total workload of all surgeries from both instances.

We evaluate the ε-proximity quantity (a_{ij}) for every combination of instances (i and j). When all instances are generated and compared, we select the X instances among which the maximum proximity (Z) is minimal. We do this by solving the following ILP. Alternatively, the Ford-Fulkerson algorithm or Hopcroft-Karp algorithm could be used.

$$
\begin{aligned}
&\min Z \\
&\sum_i Y_i = X \\
&Q_{ij} \leq Y_i \ (\forall i, j) \\
&Q_{ij} \leq Y_j \ (\forall i, j) \\
&Q_{ij} \geq Y_i + Y_j - 1 \ (\forall i, j) \\
&a_{ij} Q_{ij} \leq Z \ (\forall i, j) \\
&Q_{ij}, Y_i \varepsilon \{0, 1\}, Z \geq 0
\end{aligned}
$$

Here, binary variable Y_i indicates whether we select instance i, binary variable $Q_{i,j}$ is 1 iff both $Y_i = 1$ and $Y_j = 1$ (i.e., instances i and j are both selected). These selected X instances are thus the least ε-proximal within the original instances set, and therefore the X most different instances. In "Appendix I" we show some statistics of this procedure, and we summarize this proximity maximization procedure in Box 2.

5.4 Benchmark set

The benchmark set satisfies all four conditions of Vanhoucke and Maenhout (2007): Diversity, realism, size, and extendibility. The instances are based on a wide range of case mix types, either based on real-life case mixes from one of 11 surgical specialties, or based on one of 17 theoretical case mix profiles. In addition, $|J| * |A| = 80$ variations of problem size characteristics are used, as shown in Tables 1 and 3. As a result, the benchmark set is very *diverse*. This is further strengthened by our instance generation procedure, in

Box 2 Instance generation procedure

Repeat this procedure for each parameter combination

Step 1: Generate $3X$ random instances.

Step 2: Determine for all surgery pairs of all pairs of instances if they are ε-proximal.

Step 3: Determine the maximum matching of ε-proximate surgeries.

Step 4: Determine the ε-proximity quantity of all pairs of instances.

Step 5: Use the ILP to select X instances for which the maximum ε-proximity quantity between all instance pairs is minimal.

which we use the concept of instance proximity to maximize instance diversity. In "Appendix II" we show some statistics on the case mix diversity of the generated instances.

The benchmark set reflects *real-world* problems by the use of the underlying patient type dataset. The instances differ in *size*, to facilitate a benchmark set with a mix of smaller and larger instances for solvability reasons. Other researchers can *extend* the benchmark set with their own characteristics. Also, the instance generator is provided on the website, for the reader to generate even more instances.

To enable researchers to assess a solution method's performance using a smaller set, we selected a subset from the total benchmark set of 22,400 instances. This smaller benchmark set consists of those instances that we were not able to solve to optimality within 10 min using CPLEX for the two variants of the surgery scheduling problem introduced in "Appendix III". Furthermore, in this subset, one instance per parameter combination was selected, and only instances of average loads (0.95, 1.0 and 1.05) are considered.

The benchmark set, the benchmark subset, the instance solution validator, our first solutions, and the instance generator are available for the academic community at:

https://www.utwente.nl/choir/en/research/BenchmarkORScheduling/

Instance and solution files are in the plain ASCII text format (tab separated), and detailed descriptions of their formats are provided on the website. To facilitate statistical and sensitivity analyses based on the benchmark set, a sample generator is also provided at this URL. For each surgery in each instance, durations can be sampled from the 3-parameter lognormal distribution of the surgery type's distribution. These samples can be used for simulation purposes.

"Appendix III" gives more details on the specifics of the surgery scheduling problem and the first solutions that we provided on the website.

All programs were developed in the Embarcadero® Delphi XE8 programming language, and compiled to MS Windows executables.

6 Conclusions and recommendations

Benchmarking the performance of surgery scheduling between different hospitals is difficult, as case mixes differ between hospitals. We have proposed a benchmark set and a case mix classification to facilitate (benchmarking) experiments. We have also developed a novel instance generation procedure that maximizes the difference between instances.

The proposed generic benchmark set for surgery scheduling algorithms is diverse, derived from real-world data, varies in size, and is extendable, according to the characteristics of an effective benchmark set (Vanhoucke and Maenhout 2007). The benchmark set, the instance generator, and solution validator can be downloaded from the website:

https://www.utwente.nl/choir/en/research/BenchmarkORScheduling/

On the website we also provide a small benchmark set consisting of a subset of hard instances of the large benchmark set. We also provide our initial solutions to all instances, and a sample generator.

We found that the diversity in the real-life case mixes is much higher than for generated (theoretical) data found in the literature. Therefore, further research is needed to analyze the relation between hospital case mixes, and case mixes used in literature. Furthermore, suitable algorithms for instances with specific case mixes can be developed. For example, we already mentioned that robust approaches are more suitable for solving instances with an underlying case mix with a high coefficient of variability.

The proposed case mix classification is visual, and gives insight into what type of case mix is under consideration. Practitioners can use the case mix classification to get insight in to what extent another case mix (than their own) is comparable to their own. Hence, given, for example, instances or case mixes found in the literature, a practitioner can assess to what extent the results are applicable in their own situation.

Acknowledgements This research is funded by NWO, Grant No. 406-14-128. The authors acknowledge the anonymous reviewers for their constructive comments and suggestions for improvements.

Appendix I: Instance proximity statistics

An analysis of the proximity statistics of the instances in the benchmark set shows that the maximum proximity of instances is influenced by the case mix and the problem size (number of ORs). Figures 13 and 14 show proximity statistics of theoretical case mix profile 2 and 3, respectively. The X-value of each dot represents the instance parameter combination. The Y-value is the maximum proximity for 10 instances that were selected from 30 random instances using the ILP-procedure in Sect. 5.3.

The figures show that the instances from theoretical case mix profile 2 are less proximal than those of theoretical case mix profile 3. This can be explained by the range of the possible distributions that can be selected in a case mix. In theoretical case mix profile 3, all distributions have a small σ and small μ, whereas in theoretical case mix profile 2, the σ and μ are both larger. Furthermore, in theoretical case mix profile 3, only small shift values can be derived, whereas in theoretical case mix profile 2 both larger and smaller shift values can occur.

As would be expected, the figures also show that a higher number of ORs affects the maximum proximity. More ORs, means more surgeries, and thus a higher possibility of drawing two similarly distributed surgeries.

The proximity statistics of all instance sets in the benchmark show similar trends. In the benchmark set we have included the detailed proximity statistics for all these instance sets (i.e., for each case mix profile and each parameter combination).

Appendix II: Benchmark instance classification

To show the case mix diversity of the benchmark instances, we plotted several generated instances based on specialty

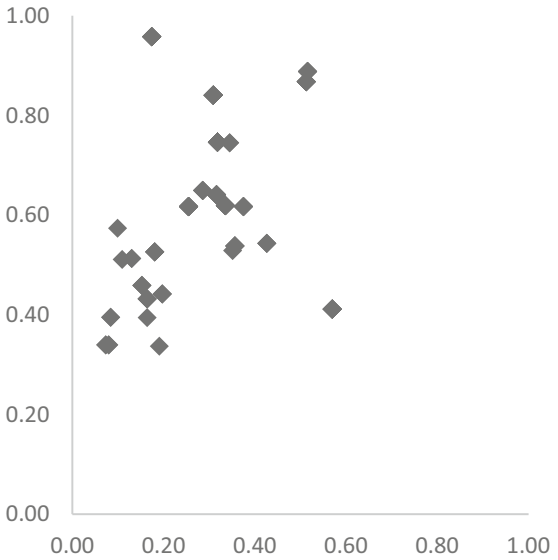

Fig. 15 Instance plot of ONC instance

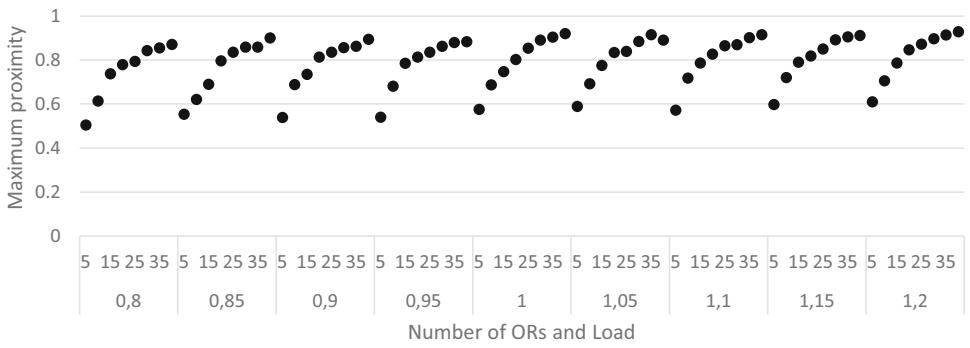

Fig. 13 Maximum proximity of selected instances of case mix type 2 in benchmark set

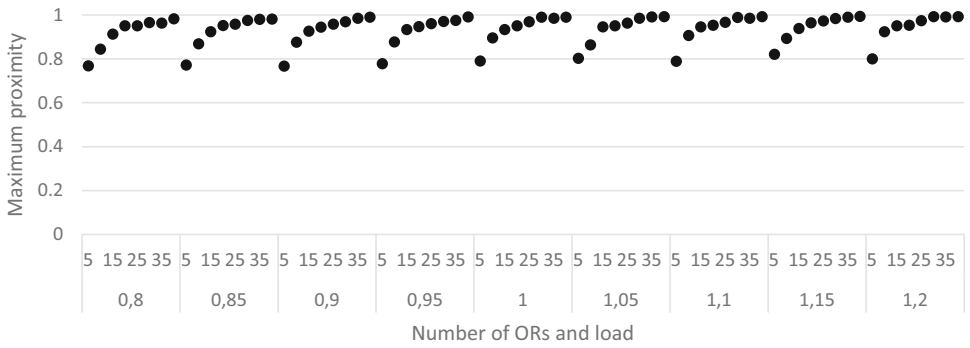

Fig. 14 Maximum proximity of selected instances of case mix type 3 in benchmark set

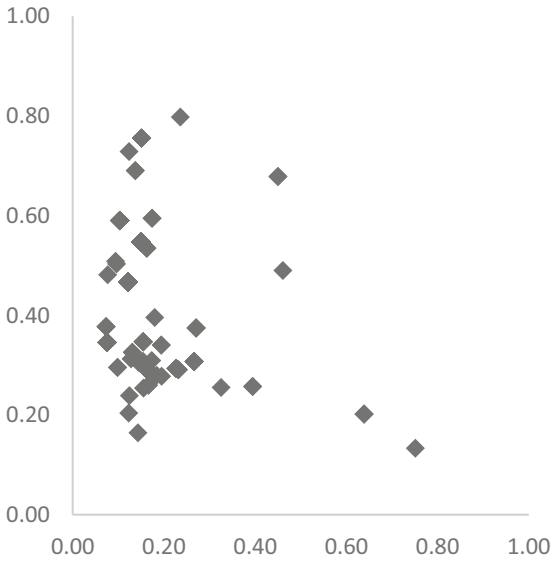

Fig. 16 Instance plot of URO instance

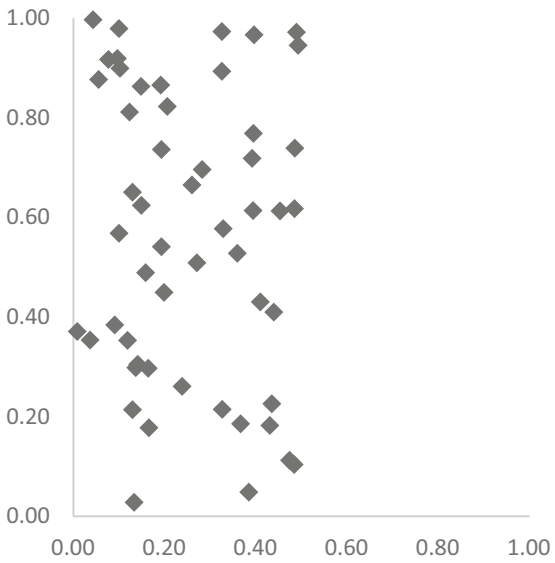

Fig. 17 Instance plot of theoretical instance no. 5

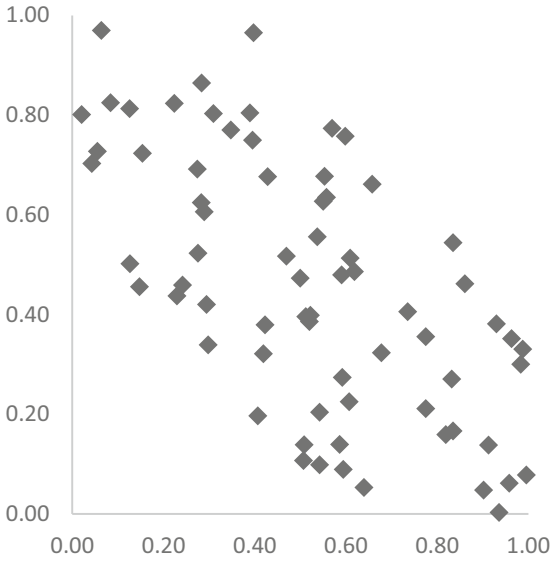

Fig. 18 Instance plot of theoretical instance no. 13

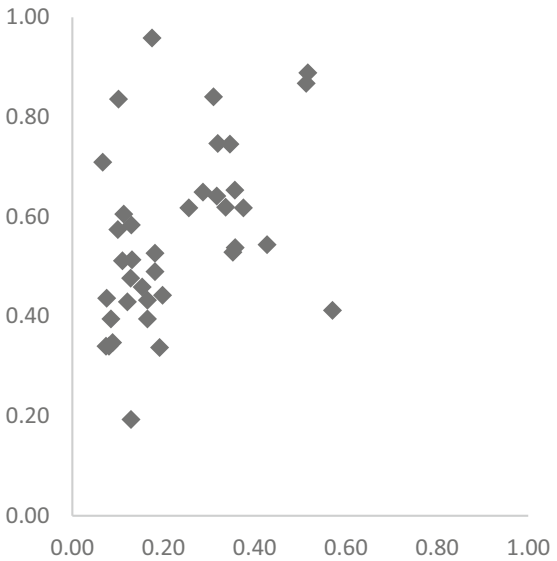

Fig. 19 Underlying case mix ONC

case mixes in the instance classification plot (see Figs. 15, 16). Furthermore, we plotted several instances based on theoretical case mix profiles (see Figs. 17, 18).

For the theoretical instances, one can easily detect the corresponding case mix profile underlying the instance. For example, Fig. 17 was based on case mix profile 5, with surgery types in the left half, whereas Fig. 18 was based on case mix profile 13. The real-life instances are harder to differentiate. Figure 19 shows the underlying specialty case mix of the instance of Fig. 15. Now, one can see that the instance corresponds with its underlying case mix.

Appendix III: Surgery scheduling problem formulation

The large amount of surgery scheduling literature consists of many problem variants. To provide some first solutions to the benchmark set developed in this paper, we present two basic surgery scheduling problem variants. Both variants consider the idle time of an operating room as performance indicator, together with a variant specific performance measure.

Variant A Planning surgeries in overtime, also known as overbooking, is not allowed. Therefore, the sum of the expected duration of all canceled patients is an important performance measure to this variant.

Variant B All surgeries need to be scheduled, if necessary in overtime. Therefore, the planned overtime is an important performance measure to this variant.

AIII.1 Formal problem formulation

Before we define the problem, we first introduce some notation, as shown in Table 4. Let $s \in S$ be the set of surgeries, and $j \in J$ the set of operating rooms. c_j is the capacity of operating room j, and $X_{s,j}$ is a binary variable that indicates whether surgery s is scheduled in operating room j. Recall that μ_s is the expected duration of surgery s.

To be able to identify an unscheduled surgery, we introduce Y_s, which is a binary variable indicating whether a surgery s is unplanned ($Y_s = 1$) or planned ($Y_s = 0$).

We want to minimize the idle time per operating room (I_j), and depending on the problem variant the expected duration of all canceled patients (C) or the planned overtime per operating room (O_j). Note that for variant A holds that $O_j = 0$, and for variant B holds that $C = 0$. This gives the following objective:

$$\min C + \sum_j \left(I_j + O_j\right)$$

We identify the following constraints:

$$\sum_j X_{s,j} + Y_s = 1 \forall s \in S$$

This constraint forces each surgery to be planned at an operating room, or be canceled.

We define the idle time, number of cancellations, and the overtime as follows:

$$I_j = \left[c_j - \sum_s X_{s,j}\mu_s\right]^+ \forall j \in J$$
$$C = \sum_s Y_s \mu_s$$
$$O_j = \left[\sum_s X_{s,j}\mu_s - c_j\right]^+ \forall j \in J$$

The idle time of an operating room is the difference between the capacity of the operating room and the sum of the scheduled durations at the operating room. The expected duration of all canceled patients equals the sum of all patients that are unscheduled times their expected duration. The overtime is the difference between the sum of the scheduled durations at the operating room and the capacity of the operating room. This gives the following constraint:

$$\sum_s X_{s,j}\mu_s = c_j + O_j - I_j \quad \forall j \in J$$

To distinguish between variants A and B, we add variant specific constraints. To ensure no patients are scheduled in overtime in variant A, we add the following constraint to the problem in variant A:

$$O_j = 0 \quad \forall j \in J$$

An operating room cannot have patients scheduled in overtime.

To ensure all patients are scheduled in variant B, we add the following constraint to the problem in variant B:

$$Y_s = 0 \quad \forall s \in S$$

As no patients are allowed to be canceled, all patients are forced to be assigned to an operating room.

Finally, we have some non-negativity constraints:

$$X_{s,j} \in \{0, 1\}, Y_s \in \{0, 1\}, I_j \geq 0, C \geq 0, O_j \geq 0$$
$$\forall s \in S, j \in J.$$

Table 4 Notation

Set, parameter or variable	Definition
$s \in S$	Set of surgeries
$j \in J$	Set of operating rooms
c_j	Capacity of operating room j
μ_s	Expected duration of surgery s
$X_{s,j}$	Binary variable indicating whether surgery s is scheduled in operating room j (1) or not (0)
Y_s	Binary variable indicating whether a surgery is unplanned (1) or planned (0)
I_s	Idle time of operating room j
C	Sum of the expected durations of all canceled surgeries
O_s	Expected overtime of operating room j

AIII.2 Solution method

To find first solutions to the surgery scheduling problem variants A and B for the benchmark instances presented in this paper, we apply the well-known list scheduling heuristic with multiple machine and job selection rules. This heuristic has been widely used in the literature for surgery scheduling (Molina-Pariente et al. 2015). The machine selection rules are best fit (BF), first fit (FF), random fit (RF), and worst fit (WF). The job selection rules are ascending order of expected surgery duration (Asc), descending order of expected surgery duration (Des), and random selection (Rnd). This yields 12 list scheduling variants, which we denote by Dur_Asc_BF, …, Dur_Rnd_WF.

References

Agrawal, M., Elmaghraby, S., & Herroelen, W. (1996). DAGEN: A generator of testsets for project activity nets. *European Journal of Operational Research*, *90*, 376–382.

Bilgin, B., Demeester, P., Misir, M., Vancroonenburg, W., & Berghe, G. V. (2012). One hyper-heuristic approach to two timetabling problems in health care. *Journal of Heuristics*, *18*, 401–434.

Brailsford, S., Harper, P., Patel, B., & Pitt, M. (2009). An analysis of the academic literature on simulation and modelling in health care. *Journal of Simulation*, *3*, 130–140.

Brailsford, S., & Vissers, J. (2011). OR in healthcare: A European perspective. *European Journal of Operational Research*, *212*, 223–234.

Brucker, P., Burke, E. K., Curtois, T., Qu, R., & Berghe, G. V. (2010). A shift sequence based approach for nurse scheduling and a new benchmark dataset. *Journal of Heuristics*, *16*, 559–573.

Burke, E. K., De Causmaecker, P., Berghe, G. V., & Van Landeghem, H. (2004). The state of the art of nurse rostering. *Journal of Scheduling*, *7*, 441–499.

Cardoen, B., & Demeulemeester, E. (2011). Operating room planning and scheduling: A classification scheme. *International Journal of Health Management and Information*, *1*, 71–83.

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

Cayirli, T., & Veral, E. (2003). Outpatient scheduling in health care: A review of literature. *Production and Operations Management*, *12*, 519–549.

Ceschia, S., Dang, N. T. T., De Causmaecker, P., Haspeslagh, S., & Schaerf, A. (2015). Second international nurse rostering competition (INRC-II)—problem description and rules. arXiv:1501.04177.

Ceschia, S., & Schaerf, A. (2016). Dynamic patient admission scheduling with operating room constraints, flexible horizons, and patient delays. *Journal of Scheduling*, *19*, 377–389.

Curtois, T. (2016). Employee shift scheduling benchmark data sets. http://www.cs.nott.ac.uk/~psztc/NRP/. Accessed November 22, 2016.

Demeester, P., Souffriau, W., De Causmaecker, P., & Berghe, G. V. (2010). A hybrid tabu search algorithm for automatically assigning patients to beds. *Artificial Intelligence in Medicine*, *48*, 61–70.

Demeulemeester, E., Vanhoucke, M., & Herroelen, W. (2003). RanGen: A random network generator for activity-on-the-node networks. *Journal of Scheduling*, *6*, 17–38.

Demirkol, E., Mehta, S., & Uzsoy, R. (1998). Benchmarks for shop scheduling problems. *European Journal of Operational Research*, *109*, 137–141.

Denton, B., Viapiano, J., & Vogl, A. (2007). Optimization of surgery sequencing and scheduling decisions under uncertainty. *Health Care Management Science*, *10*, 13–24.

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

Drexl, A., Nissen, R., Patterson, J. H., & Salewski, F. (2000). ProGen/πx—An instance generator for resource-constrained project scheduling problems with partially renewable resources and further extensions. *European Journal of Operational Research*, *125*, 59–72.

Gehring, H., & Homberger, J. (2001). A parallel two-phase metaheuristic for routing problems with time windows. *Asia-Pacific Journal of Operational Research*, *18*, 35–47.

Hans, E., Wullink, G., Van Houdenhoven, M., & Kazemier, G. (2008). Robust surgery loading. *European Journal of Operational Research*, *185*, 1038–1050.

Haspeslagh, S., De Causmaecker, P., Schaerf, A., & Stølevik, M. (2014). The first international nurse rostering competition 2010. *Annals of Operations Research*, *218*, 221–236.

Homberger, J. (2012). A (μ, λ)-coordination mechanism for agent-based multi-project scheduling. *OR Spectrum*, *34*, 107–132.

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

Kok, A. L., Meyer, C. M., Kopfer, H., & Schutten, J. M. J. (2010). A dynamic programming heuristic for the vehicle routing problem with time windows and European Community social legislation. *Transportation Science*, *44*, 442–454.

Kolisch, R., Schwindt, C., & Sprecher, A. (1999). Benchmark instances for project scheduling problems. In *Project scheduling* (pp. 197–212). Dordrecht: Kluwer

Kolisch, R., & Sprecher, A. (1996). PSPLIB—A project scheduling problem library: OR software-ORSEP operations research software exchange program. *European Journal of Operational Research*, *96*, 205–216.

Kolisch, R., Sprecher, A., & Drexl, A. (1995). Characterization and generation of a general class of resource-constrained project scheduling problems. *Management Science*, *41*, 1693–1703.

Lamiri, M., Grimaud, F., & Xie, X. (2009). Optimization methods for a stochastic surgery planning problem. *International Journal of Production Economics*, *120*, 400–410.

Mannino, M., Nilssen E.J., Nordlander, T.E. (2014). SINTEF ICT: MSS-Adjusts Surgery data. https://www.sintef.no/Projectweb/Health-care-optimization/Testbed. Accessed 9 March 2015.

Marcon, E., Kharraja, S., & Simonnet, G. (2003). The operating theatre planning by the follow-up of the risk of no realization. *International Journal of Production Economics*, *85*, 83–90.

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

May, J. H., Strum, D. P., & Vargas, L. G. (2000). Fitting the lognormal distribution to surgical procedure times. *Decision Sciences*, *31*, 129–148.

Molina-Pariente, J. M., Hans, E. W., Framinan, J. M., & Gomez-Cia, T. (2015). New heuristics for planning operating rooms. *Computers and Industrial Engineering*, *90*, 429–443.

Musliu, N., Schaerf, A., & Slany, W. (2004). Local search for shift design. *European Journal of Operational Research*, *153*, 51–64.

Pillac, V., Gueret, C., & Medaglia, A. L. (2013). A parallel matheuristic for the technician routing and scheduling problem. *Optimization Letters*, *7*, 1525–1535.

Qu, R., Burke, E. K., McCollum, B., Merlot, L. T., & Lee, S. Y. (2009). A survey of search methodologies and automated system development for examination timetabling. *Journal of Scheduling*, *12*, 55–89.

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

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, *35*, 254–265.

Stepaniak, P. S., Heij, C., Mannaerts, G. H., de Quelerij, M., & de Vries, G. (2009). Modeling procedure and surgical times for current procedural terminology-anesthesia-surgeon combinations and evaluation in terms of case-duration prediction and operating room efficiency: A multicenter study. *Anesthesia and Analgesia*, *109*, 1232–1245.

Tyler, D. C., Pasquariello, C. A., & Chen, C.-H. (2003). Determining optimum operating room utilization. *Anesthesia and Analgesia*, *96*, 1114–1121.

Van Riet, C., & Demeulemeester, E. (2014). Trade-offs in operating room planning for electives and emergencies: A review. SSRN 2553849.

Vanhoucke, M., & Maenhout, B. (2007). NSPLib—A nurse scheduling problem library: a tool to evaluate (meta-) heuristic procedures. In *Operational research for health policy: making better decisions, proceedings of the 31st annual meeting of the working group on operations research applied to health services* (pp. 151–165).

Vanhoucke, M., & Maenhout, B. (2009). On the characterization and generation of nurse scheduling problem instances. *European Journal of Operational Research*, *196*, 457–467.

Wauters, T., Kinable, J., Smet, P., Vancroonenburg, W., Berghe, G.V., & Verstichel, J. (2016). The multi-mode resource-constrained multi-project scheduling problem. *Journal of Scheduling*, *19*(3), 271–283.