

# An efficient computational method for large scale surgery scheduling problems with chance constraints

Mahdi Noorizadegan<sup>1</sup> · Abbas Seifi<sup>1</sup>

Received: 31 October 2016 / Published online: 4 October 2017  
© Springer Science+Business Media, LLC 2017

**Abstract** We propose an efficient solution method based on a decomposition of set-partitioning formulation of an integrated surgery planning and scheduling problem with chance constraints. The studied problem is characterized by a set of identical operating rooms (ORs), a set of surgeries with uncertain durations, a set of surgeons, and surgery dependent turnover times. The decision variables include the number of ORs to open, assignments of surgeries and surgeons to ORs in admissible periods, and the sequence of surgeries to be performed in a period. The objective is to minimize the cost of opening ORs and the penalties associated with turnover times. In the proposed formulation, the column generation subproblem is decomposed over ORs and time periods. The structure of the subproblem is further exploited and transformed to a shortest path problem. A search algorithm has been devised to efficiently solve the resulting subproblem, subject to some optimality and feasibility conditions. The proposed computational method outperforms the standard chance constrained model and reduces the solution time significantly. Furthermore, extensive simulation experiments have been carried out to compare the performance of three variants of the underlying formulations and evaluate the sensitivity of the decisions to the probability of exceeding a session length.

**Keywords** Surgery planning and scheduling · Set-partitioning formulation · Integer chance constrained programming · Integer programming

---

This research was partially funded by Iran's National Elites Foundation.

---

✉ Abbas Seifi  
aseifi@aut.ac.ir

Mahdi Noorizadegan  
m.noorizadegan@aut.ac.ir

<sup>1</sup> Industrial Engineering Department, Amirkabir University of Technology, Tehran, Iran

## 1 Introduction

Studying surgery planning and scheduling problems is motivated by their significant contributions to hospitals revenue. Surgical services account for more than 40% of hospitals revenue and cost and this figure is expected to grow in near future [10]. Therefore, maximizing operating rooms utilizations and reducing the cost of surgery operations has gain considerable attention in recent years.

A surgery planning problem may be considered as a composition of two phases: (a) an assignment phase which consists of assigning patients and surgeons to operating rooms (ORs) in certain time periods (e.g., days or shifts), and (b) a scheduling phase that is to find the best sequence of surgeries for each OR in the time period determined in the assignment phase. Due to the complexity of integrating these two phases, a majority of studies have considered them separately [5]. A common approach is to assign surgeries based on a *block surgery policy*. In this policy, each OR available time is divided into a certain number of blocks. Each block is then allocated to one surgery type. Once the block allocations are determined, surgeries of each type are assigned to the allotted blocks. Despite the efficiency of disaggregating the two problems, which allows one to solve larger instances, the solutions are not guaranteed to be optimal and may in fact lead to hospital operational inefficiency [29]. Therefore, our motivation is to integrate both phases and solve a large scale surgery planning and scheduling problem together.

Historically, the main focus of the literature has been on deterministic OR scheduling problems [22] while assumptions on parameters with random nature such as surgery durations have a significant impact on both availability and quality of services. For example, when the uncertain nature of surgery duration is ignored, the patients' and surgeons' waiting time and OR overtime may increase. Stochastic OR scheduling problems are commonly studied by considering two sources of uncertainty: in parameters related to duration (e.g., surgery duration) and in patient's arrival time. Stochastic optimization and robust optimization methods [16, 19] have been used to deal with the pertinent uncertainty in surgery scheduling problems. We present a brief review on some relevant studies next. Cardoen et al. [5] and Erdogan and Denton [9] provide extensive reviews on different aspects of OR planning and scheduling problems.

Zhang and Xie [30] studied a dynamic OR assignment problem in which surgery durations were uncertain and surgeons were assigned to available ORs on a first-come first-serve basis. They used a simulation-based optimization approach to solve the problem.

Neyshabouri and Berg [19] address a surgery scheduling problem where elective surgeries are assigned to a set of available surgery blocks. They also include downstream decisions (post-operation care) in their formulation. They assume that surgery durations and lengths of stay are uncertain, and use a column generation method to solve their proposed model. Denton et al. [8] propose a two-stage robust optimization model for a surgery problem in which the overtime is penalized. They find bounds for the number of ORs and the number of uncertain parameters i.e., surgery durations that can reach their upper bounds.

Gul et al. [12] study a surgery allocation problem under uncertainty over a finite planning horizon. Their objective function is to minimize cancellation, waiting, and

overtime costs. Using a progressive hedging algorithm for their multi-stage stochastic mixed-integer formulation, they can solve relatively large instances (up to 50 surgeries and 10 days) and find near optimal surgery schedules. In addition to robust optimization based model, Denton et al. [8] consider a surgery block assignment problem with uncertain surgery durations to minimize OR opening and overtime costs. They propose a two-stage stochastic optimization formulation as well as a robust optimization formulation and solve the resulting problems using a cutting plane method and some heuristics. Wang et al. [28] extend [8] by considering a sample average approximation method to transform a two-stage stochastic model to a deterministic model for the surgery planning problem. They solve the resulting problem using a column generation method. Batun et al. [4] also propose a two-stage mixed integer stochastic model for a surgery assignment problem to minimize ORs opening and overtime costs. They aim to find the optimal assignment of pairs of surgery-surgeon to ORs on a daily basis. Shylo et al. [25] study a batch scheduling problem within a block booking policy and maximize the expected utilization of ORs while some probabilistic capacity constraints are imposed over ORs' availability time.

Due to the complexity of OR scheduling problems, the majority of the literature focuses on only one of the two phases (e.g., the block surgery policy) with relatively restrictive settings such as a single-period problems or ignoring surgeons in OR scheduling. The surgery planning and scheduling problems with random parameters may be formulated as stochastic integer programs (SIPs) which are among the most challenging problems in operations research [24]. The efficiency of solution algorithms for this class of problems highly depends on clever application of integer programming methods to solve SIP. To this end, one could study the possibility of taking advantage of the problem structure to develop a tailor-made algorithm which has not been well investigated for SIP in the literature. In particular, Chance Constrained Programming (CCP) is still computationally challenging [14] although its development dates back to the 1960s. There exist two main approaches to invoking uncertainty in an integer chance constrained program: a discrete approximation of uncertainty by scenarios [14] or a continuous approximation of the distribution functions of uncertain parameters [18]. Song et al. [26] provide a thorough discussion on the challenges and limitations of these two formulations. The former results in a huge number of binary variables and the latter leads to a non-linear integer program that is only tractable when random parameters follow a normal distribution. Assuming normal distribution would resolve the nonconvexity of chance constraints and makes the continuous approximation tractable. However, there are many situations where surgery durations are not normally distributed. In such situations, one could resort to discrete approximation. Here, we solve the problem using both approaches to show the impact of such approximations on the final variables.

In the literature of SIP, branch-and-cut methods are widely used to solve SIP [1, 23]. Although branch-and-price methods are among efficient and popular methods for integer programs [2, 3], very few papers have investigated solving SIP using branch-and-price methods. Lulli and Sen [15] use a set-partitioning formulation and a branch-and-price method for a chance constrained formulation of a batch sizing problem. Their proposed method could not outperform the branch-and-cut method in many instances due to the complexity of solving the subproblems. Therefore, an interesting research

question arises that whether SIPs can be efficiently solved using a branch-and-price method.

In this paper, we address this research question by focusing on a multi-attribute chance constrained formulation of an integrated surgery planning and scheduling problem. We use a set-partitioning formulation of the problem, and extend the branch-and-price method in [20] to efficiently solve it. The problem is characterized by a set of identical operating rooms (ORs), a set of surgeries with uncertain surgery durations, a set of surgeons, and surgery dependent turnover times. The here-and-now decisions include the number of ORs to open, assignments of surgeries and surgeons to ORs in some admissible (preferred) time periods, and the sequence of surgeries to be performed in a period. We do not consider emergency surgeries. The turnover time is penalized in order to increase OR utilization and control/reduce surgeons waiting times. The objective function is to minimize the cost of opening ORs and the penalties associated with turnover times.

Four additional realistic conditions are taken into account. (1) The number of available ORs in each period is limited. (2) Each surgeon has his/her own surgery list. As a result, we assume that each surgeon can be assigned to only one OR in any time period. (3) Each surgery has a set of pre-defined admissible or preferred periods and cannot be scheduled in other time periods. This is a common practice in hospital management and is due to the availability of surgeons or preferences of the patients. (4) We consider surgery dependent turnover time that consists of the time required for post-surgery activities (including cleaning) and setting up the room for next surgery. We assume that induction surgical and awakening activities are included in surgery duration. However, one could model these activities differently. The turnover time could depend on the preceding surgery or the position of a surgery in the sequence. For example, after a patient's surgery with a transmissible disease, the OR must be carefully cleaned up before the next surgery could be performed in that room. Note that our focus in this research is on assigning surgeons and surgery sessions to ORs and time periods. Additional resources such as anaesthetists and nurses have not been taken into consideration.

The contributions of this paper is summarized as follows:

- We propose an efficient solution method for an integrated surgery planning and scheduling problem with uncertain surgery durations. The proposed method can also be adapted for other general SIPs. The problem is formulated as a set-partitioning model and solved using a customized branch-and-price method.
- The column generation subproblem is decomposed over ORs and time periods. We exploit the structure of the column generation subproblem and reformulate it as a shortest path problem. The resulting shortest path is then solved using a search algorithm satisfying *optimality and feasibility conditions*. The proposed method leads to a significant reduction in solution time comparing with that of the standard chance constrained model. In comparison to the largest problems reported in the literature, we have solved relatively larger instances (up to 60 surgeries, 5 periods, 10 ORs and 13 surgeons) with the above realistic assumptions to optimality.

- The chance constrained formulation of OR session lengths are written for two types of uncertainties represented by: (a) a set of discrete scenarios and (b) the probability density function of the uncertain parameters.
- We have also carried out an extensive simulation to study the impact of the probability of exceeding OR session length on the decision variables and the objective function. This analysis helps hospitals' management to decide on a proper service level for ORs session lengths and provides them with useful managerial insights.

The remainder of the paper is structured as follows. We introduce the set-partitioning model and the column generation subproblem in Sect. 2. The column generation subproblem is reformulated in Sect. 3. We outline the steps of the proposed search algorithm in Sect. 4. Computational results and sensitivity analysis are presented in Sect. 5. Some concluding remarks and future research are given in Sect. 6.

## 2 The set-partitioning model

We propose a set-partitioning formulation for the described integrated surgery planning and scheduling problem. We use chance constrained programming to handle uncertainty i.e., the total surgery time (including the turnover time) of an assignment must be within the OR session length with a pre-specified probability  $(1 - \epsilon)$ .

Let  $I$ ,  $K$  and  $R$  be the sets of surgeries, surgeons and identical ORs. The surgery list of surgeon  $k$ , denoted by  $I_k$ , defines the set of surgeries to be performed by the surgeon on the planning horizon ( $T$ ). We consider a set of admissible periods for each surgery ( $T_i$ ,  $i \in I$ ). Also let  $L$  be the normal session length. The uncertain duration of surgery  $i$  and its mean are denoted by  $\bar{d}_i$  and  $\bar{d}_i$ , respectively. The uncertainty of surgery durations is presented by either a set of scenarios ( $\mathcal{S}$ ) where  $p_s$  is the probability assigned to scenario  $s$ , or by distribution functions  $\mathcal{D}(\bar{d}_i, \sigma_i^2)$  where  $\sigma_i^2$  is the variance of surgery  $i$ .

Let  $v_{it}$  be a binary variable that takes a value of 1 if surgery  $i$  is assigned to an OR in period  $t$ , and zero otherwise. A sequence is defined as an ordered subset of surgeries ( $I_s$ ) assigned to an OR in a period and is considered feasible if

- (a) its total time satisfies the chance constraint condition,
- (b) it is assigned to a time period which belongs to the intersection of the admissible periods of all surgeries in that sequence ( $t \in \{\cap_{i \in I_s} T_i\}$ ),
- (c) there is no gap between two consecutive surgeries in the sequence, i.e., ORs are not left idle.

We denote sequence  $s$  by binary variable  $u_s$ . It takes a value of 1 if it is chosen in the optimal solution. The cost of sequence  $s$  is denoted by  $c_s$  and consists of the fixed opening cost and the penalized turnover time for all surgeries in the sequence. Also,  $S_t$  and  $S_t(i)$  in a time period  $t \in T$  are the sets of all feasible sequences, and the set of all feasible sequences that include surgery  $i$  ( $S_t(i) = \{s \in S_t | i \in I_s\}$ ), respectively. The set of feasible sequences at period  $t$  in which surgeon  $k$  has at least one surgery is denoted by  $S_t^k$ .

The surgery duration is assumed to follow a known distribution function (continuous or discrete). W.L.O.G., we can assume the duration of the pre-surgery ( $pre_i$ ) and

the post-surgery ( $post_i$ ) activities are not included in the surgery duration and are separately considered in the model. As these activities are common, and the variance of their durations (individually) are very small, we can assume  $pre_i$  and  $post_i$  are exactly known in advance. Furthermore in real cases, the pre-surgery and post-surgery time may depend on the previous surgery and/or the next surgery. To simplify the notation, we summarize them in a square matrix and denote its elements by  $\tau_{ij}$ ,  $i, j \in I$  which is the duration of activities required for switching from surgery  $i$  to surgery  $j$  i.e.,  $\tau_{ij} = post_{ij} + pre_{ij}$ . The entries on the diagonal of the matrix are the duration of the pre-surgery time if surgery  $i$  is the first surgery in the sequence. The post-surgery time of surgery  $i$  is denoted by  $\bar{\tau}_i$  when it is the last surgery of the sequence which could include a complete cleaning. It is clear that one can include the preparation activities in the surgery duration. We also assume that surgeons are not assigned to next surgeries during the turnover time.

The set-partitioning model for the surgery scheduling problem is then formulated as:

$$P: \min \sum_{t \in T} \sum_{s \in S_t} c_s u_s \tag{1a}$$

$$\text{s.t. } \sum_{t \in T_i} v_{it} = 1, \forall i \in I \tag{1b}$$

$$v_{it} - \sum_{s \in S_t(i)} u_s \leq 0, \forall t \in T_i, \forall i \in I \tag{1c}$$

$$\sum_{s \in S_t^k} u_s \leq 1, \forall k \in K, \forall t \in T \tag{1d}$$

$$\sum_{s \in S_t} u_s \leq m, \forall t \in T \tag{1e}$$

$$u_s \in \{0, 1\}, \forall s \in S_t, \forall t \in T \tag{1f}$$

$$v_{it} \in \{0, 1\}, \forall i \in I, \forall t \in T. \tag{1g}$$

The objective function computes the total cost including the fixed opening cost and the penalty associated with the turnover time between surgeries i.e.,  $c_s = c_R + \sum_{i,j \in I_s} c_{ij} \tau_{ij} + c_i \bar{\tau}_i$  where  $c_R$ ,  $c_{ij}$  and  $c_j$  are the fixed opening cost, the unit penalty cost of turnover time of switching from surgery  $i$  to surgery  $j$  and surgery  $i$  to the end of session, respectively. Constraint (1b) ensures that each surgery is assigned to one period of its admissible periods. Constraint (1c) guarantees that at least one schedule contains surgery  $i$  for an admissible period. Constraint (1d) makes sure that a surgeon can be assigned to at most one OR in each period. Constraint (1e) enforces the limitation on the number of available ORs in each period such that  $m = |R|$ . Constraints (1f–1g) present the integrality conditions on variables.

It is impractical to include all feasible sequences in the beginning to P. We adopt the following policy which was successfully used by [11,21]. Starting with a problem called the *Restricted Master Problem* (RMP), a subset of feasible sequences is considered in the beginning. Feasible sequences which improve the current solution are

iteratively identified and added to RMP. This policy is implemented within a branch-and-bound algorithm which leads to a *branch-and-price method*. The subproblem which identifies feasible and improving sequences is known as the *column generation subproblem* or *pricing subproblem*. The column generation subproblem is constructed based on the dual problem of the linear relaxation of the master problem as follows.

Let  $\beta_i, \alpha_{it}, \gamma_{kt}$  and  $\lambda_t$  be the dual variables associated with constraints (1b–1e) in the linear relaxation of the master problem. The dual problem is formulated by

$$\text{DP: } \max \sum_{i \in I} \beta_i - \sum_{t \in T} \sum_{k \in K} \gamma_{kt} - m \sum_{t \in T} \lambda_t \tag{2}$$

$$\text{s.t. } \sum_{i \in I_s} \alpha_{it} - \sum_{k \in K_s} \gamma_{kt} - \lambda_t \leq c_s, \forall s \in S_t, \forall t \in T \tag{3}$$

$$\beta_i - \alpha_{it} \leq 0, \forall t \in T_i, \forall i \in I \tag{4}$$

$$\beta_i \text{ free, } \alpha_{it} \geq 0, \gamma_{kt} \geq 0 \text{ and } \lambda_t \geq 0, \forall i \in I, \forall t \in T, k \in K \tag{5}$$

where  $K_s$  is the set of surgeons assigned to sequence  $s$  at time period  $t$ , respectively. Improving sequences are those that violate constraint (3). As constraint set (3) suggests, the column generation subproblem is decomposed for each period and forms a sequence which can be assigned to an OR. In order to model the column generation subproblem for each sequence, we define the following binary decision variables for  $r \in R$  and  $t \in T$ :

- $x_{il}^{rt} = 1$  if  $i \in I$  is assigned to  $l$ th position of a sequence, and zero otherwise,
- $y_{ij}^l = 1$  if  $i \in I$  precedes  $j \in I$  in  $l$ th position of a sequence, and zero otherwise,
- $w_i^{rt} = 1$  if  $i \in I$  is the last surgery of a sequence, and zero otherwise,
- $q_k^{rt} = 1$  if surgeon  $k \in K$  is assigned a surgery from its surgery list, and zero otherwise,
- $\pi_\zeta^{rt} = 1$  if the constraint associated with scenario  $\zeta \in S$  is satisfied, and zero otherwise.

Here, we assume that the uncertainty of the surgery duration is presented by a set of scenarios  $S$ . The column generation subproblem for  $t \in T$  and  $r \in R$  is as follows:

$$z = \min \ c_r + \sum_{i,j,l \in I} c_{ij} \tau_{ij} y_{ij}^l + \sum_{i \in I} c_i \bar{\tau}_i w_i^{rt} - \sum_{i,l \in I} \alpha_{it} x_{il}^{rt} + \sum_{k \in K} \gamma_{kt} q_k^{rt} + \lambda_t$$

$$\text{s.t.: } \sum_{i \in I} x_{il}^{rt} \geq \sum_{i \in I} x_{i,l+1}^{rt}, \forall l \in I \tag{6a}$$

$$x_{il}^{rt} \leq \sum_{j \in I} x_{j,l+1}^{rt} + w_i^{rt}, \forall i, l \in I \tag{6b}$$

$$w_i^{rt} \leq \sum_{l \in I} x_{il}^{rt}, \forall i \in I \tag{6c}$$

$$x_{il}^{rt} + x_{j,l+1}^{rt} - 1 \leq y_{ij}^l, \forall i, j, l \in I \text{ and } i \neq j \tag{6d}$$

$$\sum_{i \in I_k} \sum_{l \in I} x_{il}^{rt} \leq |I| q_k^{rt}, \forall k \in K \tag{6e}$$

$$\sum_{i,l \in I} d_i^s x_{il}^{rt} + \sum_{i,j,l \in I} \tau_{ij} y_{ij}^l + \sum_{i \in I} \bar{\tau}_i w_i^{rt} \leq L + M(1 - \pi_\zeta^{rt}), \forall \zeta \in \mathcal{S} \tag{6f}$$

$$\sum_{\zeta \in \mathcal{S}} p_\zeta \pi_\zeta^{rt} \geq 1 - \epsilon \tag{6g}$$

$$x_{il}^{rt}, y_{ij}^l, w_i^{rt}, q_k^{rt}, \pi_\zeta^{rt} : \text{Binary } \forall i, l, j \in I, \zeta \in \mathcal{S} \tag{6h}$$

where  $d_i^s$  is the duration of surgery  $i$  in scenario  $s$ . The objective function is the reduced cost of a sequence consisting of the fixed opening cost, the turnover penalty and the dual variables. Constraint (6a) forbids assigning surgeries to position  $l + 1$  when no surgery is allocated to position  $l$ . Constraints (6b and 6c) ensure that  $w_i^{rt}$  takes a value of 1 if surgery  $i$  is the last surgery in the sequence. If surgery  $i$  precedes surgery  $j$  in  $l$ th position of the sequence,  $y_{ij}^l$  takes a value of 1. This is enforced by constraint (6d). Constraint (6e) states that if a surgery of surgeon  $k$  is selected, then  $q_k^{rt} = 1$ . Note that at most  $|I|$  surgeries can be assigned to an OR. Constraints (6f and 6g) impose the probability condition using the big  $M$  method. The former constraint computes the duration of the sequence consisting of the surgery durations and the surgery turnover times for each scenario. Note that  $M$  is a big positive number. The latter ensures that the total probability of the scenario constraints satisfied, is at least  $1 - \epsilon$ .

The feasibility conditions are fulfilled for a sequence if it satisfies the constraints above. The optimality condition is achieved if there exists no sequence with negative reduced cost on the entire branch-and-bound tree of the branch-and-price algorithm.

### 3 Reformulating the column generation subproblem

Although solving the above integer problem in each iteration is doable, it is not computationally very efficient. Since the aim is to generate a sequence of surgeries that violates (3), we reformulate the column generation subproblem by a shortest path problem on a graph where a sequence corresponds with a path on the graph. In order to guarantee the optimality of the solution, the optimality and feasibility conditions will be defined.

Let us assume that each surgery uniquely corresponds to one node of Graph  $G(I, E)$  where  $E$  is the set of arcs connecting nodes (surgeries). There exists an arc between node  $i$  and node  $j$  if they have at least one admissible period in common. Following constraint (3), we define the reduced cost of a sequence and the reduced cost of arc  $(i, j)$  by

$$\bar{c}_s = c_s - \sum_{i \in I_s} \alpha_{it} + \sum_{k \in K_s} \gamma_{k,t} + \lambda_t, \tag{7}$$

$$\bar{c}_{ij} = c_{ij} \frac{\tau_{ij}}{2} - \frac{\alpha_{it}}{2} - \frac{\alpha_{jt}}{2} + \gamma_{k(j)t} \mathbb{I}_{ks} \tag{8}$$

where  $\mathbb{I}_k$  is an indicator function with value of 1 if the surgeon  $k$  is scheduled in the sequence for the first time by surgery  $j$ .



*Optimality conditions* Finding a sequence of surgeries which violates constraint (3) is the same as finding a path with negative length between any pair of nodes in Graph  $G$  i.e.,

$$\bar{c}_s = \sum_{i,j \in I_s} \bar{c}_{ij} + \lambda_t + c_R < 0.$$

If such a path is identified, it will be an improving sequence.

Constrained shortest path problem is a NP-complete problem [27]. Similar shortest path problems can be found in variants of vehicle routing problems. Labelling algorithms such as Bellman-Ford algorithm and Dijkstra’s algorithm can be adopted to search the space and find improving sequences. Following the approach applied in [20] instead of one label, we use a set of labels for each node and define *dominating rules* to keep only a subset of labels.

*Feasibility conditions* Feasibility condition (a) states that a sequence is feasible if the sum of the surgery duration and the turnover time is less than or equal to the OR session length ( $L$ ) with a pre-specified probability ( $1 - \epsilon$ ). Mathematically speaking:

$$\Pr \left( \sum_{i \in I_s} \tilde{d}_i + \sum_{i,j \in I_s} \tau_{ij} + \bar{\tau}_{i^*} \leq L \right) \geq 1 - \epsilon \tag{9}$$

where  $i^*$  is the last surgery on the sequence. The closed form of the left-hand side of (9) can be written for distribution functions with the accumulative property such as normal distribution functions. A distribution function has the accumulative property if  $\tilde{d}_i, \tilde{d}_j \sim \mathcal{D}$ , then  $\tilde{d}_i + \tilde{d}_j \sim \mathcal{D}$ . Moreover, the above probability constraint can be verified for the scenario-based case. In all cases, the complexity of the problem remains tractable.

In our search algorithm, this condition can be checked for each sequence in each step. If the feasibility condition is not met, the label associated with the sequence will be omitted from the label set of that surgery even if it is not dominated. It is trivial that an infeasible sequence cannot turn into a feasible sequence in latter iterations, when new surgeries are added to the label set. This condition only prohibits infeasible sequences and does not affect the optimality condition.

Feasibility condition (b) is satisfied when the search algorithm is restricted to induced subgraph  $G_t(I_t, E_t)$  whose nodes are admissible in period  $t$  i.e.,  $I_t = \{i | t \in T_i\}$ . The resulting induced subgraph for time period  $t$  is a complete graph (a clique). Feasibility condition (c) is fulfilled by the search algorithm construction.

The optimality and feasibility conditions at each node of branch-and-bound tree is achieved when no feasible path between any pair of surgeries can be found for which total reduced cost is negative i.e.,  $\bar{c}_s < 0$ .

## 4 The search algorithm

Here, we describe the search algorithm designed based on Dijkstra's shortest path algorithm to solve the reformulated column generation subproblem. Let  $L(i)$  be a set containing the labels of node  $i$ . The  $\kappa$ th label of node  $i$ , denoted by  $L_\kappa(i)$ , is associated with a path to node  $i$  and consists of five components: the total reduced cost ( $\bar{c}_\kappa(i)$ ), the expected duration ( $d_\kappa(i)$ ), the period ( $t_\kappa(i)$ ), the operating room ( $r_\kappa(i)$ ), the nodes on the path leading to node  $i$  ( $p_\kappa(i)$ ) i.e.,  $L_\kappa(i) \leftarrow (\bar{c}_\kappa(i), d_\kappa(i), t_\kappa(i), r_\kappa(i), p_\kappa(i))$ . We denote the ordered list of all the labels by  $Q$ . Set  $Q$  is arranged on an ascending order based on the labels components respectively.

The introduction of dominating rules depends on the type and the distribution function of uncertain durations. We use the dominating rules introduced in [20]. For instance, when the random surgery duration follows a normal distribution, at each node of the graph, label A dominates label B if (i) the expected duration of the total duration of path A is less than or equal to that of path B, (ii) the variance of the total duration of path A is less than or equal to that of path B and (iii) the reduced cost of path A is at most equal to the reduced cost of path B. These conditions guarantee that label A dominates label B. The dominated labels will be removed from the label set of the node and list  $Q$ . The idea is that the dominated path is less likely to become an improving path.

The steps of our search algorithm are outlined below and presented in a pseudo code format in Algorithm 1. The algorithm starts with assigning a label to node  $i$  for each feasible OR-period pair. At the beginning, each node has  $|T_i|$  labels. Then, the first label in  $Q$  is selected and extended for each neighbour of the last node in the given sequence. Surgery  $j$  is called a neighbour of surgery  $i$  at period  $t$  if period  $t$  belongs to their admissible time periods, i.e.,  $\mathcal{N}_t(i) = \{j \in I | t \in T_j\}$ . Once a label is selected from  $Q$ , we delete it from  $Q$ .

We also use a *s-cycle-free* rule to avoid the negative cycles of size  $s$  or less. If the feasibility condition (9) and the *s-cycle free* condition are held, then the dominating rules are checked. If all conditions are satisfied, a new label associated with the new node is created and saved in the node's label set and set  $Q$ . When the total reduced cost of a sequence/path is negative, we stop and add the sequence to RMP. If no path between each pair of nodes was found, we branch on another variable on the branch-and-bound tree.

## 5 Computational experiments

In this section, we investigate the efficiency of the proposed method and also the impact of two key features of the underlying problem, namely, uncertain parameters and the objective function. Moreover, using sensitivity analysis and Monte-Carlo simulation, we study the impact of the variation of reliability level ( $\epsilon$ ) on decision variables and objective function.

In order to demonstrate the computational efficiency of the proposed method, we formulate and use a standard chance constrained model for the underlying problem presented in the appendix. The latter model is in fact an integration of Problem (6) over

**Algorithm 1:** Search Algorithm

```

begin
  for  $i \in I$  do
     $\kappa \leftarrow 0$ 
    for  $t \in T_i$  do
       $L_\kappa(i) \leftarrow (c_R + c_{ii}\tau_{ii} - \frac{\alpha_{it}}{2} + \gamma_{\kappa(i)t}, \tau_{ii} + \bar{d}_i, t, r, (i))$ 
       $Q.insertsort(L_\kappa(i))$ 
       $L(i).insert(L_\kappa(i))$ 
       $\kappa \leftarrow \kappa + 1$ 
    end
  end
  while  $Q \neq \emptyset$  do
     $tmpL \leftarrow Q.begin$ 
    remove  $tmpL$  from  $Q$ 
     $i \leftarrow tmpL$  node number
     $t \leftarrow tmpL$  assigned period
    for  $j \in \mathcal{N}'_t(i)$  do
      if feasibility (9) & s-cycle free conditions then
         $tmp_c \leftarrow \bar{c}_\kappa(i) + \bar{c}_{ij}$  and  $tmp_d \leftarrow d_\kappa(i) + \tau_{ij} + \bar{d}_j$ 
        if  $tmp_c + \lambda_t - \frac{\alpha_{jt}}{2} + c_{ij}\bar{\tau}_j < 0$ , then
          | Stop
        else if new label dominated then
          | continue
        else if  $tmp_c$  and  $tmp_d$  dominate any  $L_\kappa(j)$  then
          remove the dominated labels,
           $\kappa' \leftarrow |L(j)| + 1$ 
           $L_{\kappa'}(j) \leftarrow (\bar{c}_\kappa(i) + \bar{c}_{ij}, d_\kappa(i) + \tau_{ij} + \bar{d}_j, t, r, p_\kappa(i).insert(j), )$ 
           $Q.insertsort(L_{\kappa'}(j))$ 
           $L(j).insert(L_{\kappa'}(j))$ 
        end
      end
    end
  end
end
end

```

all time periods and ORs with binding constraints. In the standard CCP, the uncertainty is handled by a discrete approximation of random durations.

Due to complexity of handling some distribution functions such as log-normal distribution in chance constrained models, discrete and/or continuous approximations of the distribution function are commonly used in the literature [6]. It is known that surgery duration follows Log-Normal distribution [17] which is not a well-behaved distribution function. In addition to a discrete approximation of Log-Normal distribution function, we use normal distribution as the continuous approximation of distribution function of surgery duration, and then compare the results for both cases.

In terms of the objective function, two types of objective functions have been considered in the literature [7]. The first type minimizes the number of ORs to be opened while the second one takes into account some undesirable operational costs where other performance measures can be controlled. Here we study both types.

In summary, we study three cases of problems:

1. Discrete approximation of the random surgery duration with a cost based objective function,
2. Continuous approximation of the random surgery duration with a cost based objective function, and
3. Continuous approximation of the random surgery duration with an OR based objective function.

## 5.1 Data set

We used the statistical data provided by Min and Yih [17] which is obtained from an analysis of a real life data. According to our modelling requirements, some adjustments had to be made to the instances as follows. In their data set, surgeons are not included. Therefore, a set of surgeons in proportion to the number of surgeries was defined and surgeries were randomly assigned to them. We assume that each surgery appears only on one surgery list. Also, the set of admissible periods in which a surgery could be performed, was randomly generated. Min and Yih did not report the pre-surgery and post-surgery time. We used the following function to estimate the turnover time between two surgeries:  $\tau_{ij} = 0.05\bar{d}_i + 0.05\bar{d}_j + a$ , where  $a = 15$  if surgery  $i$  and surgery  $j$  are performed by the same surgeon i.e.,  $i, j \in \mathcal{L}_k$  for  $k \in K$ , and  $a = 20$  otherwise. We would add 20 minutes to the turnover time for an extra cleaning if the preceding patient had some infectious disease. The penalty cost for every minute of turnover time was considered to be 1, i.e.,  $c_\omega = 1$ . In our experiment, the probability of exceeding session length was set to 0.10 i.e.,  $1 - \epsilon = 0.90$ . We chose the rest of parameters as in [17]:  $L = 480$ ,  $c_R = 6240$ ,  $T = 480$  minutes and the number of available ORs was set to 10. Note that such parameter settings are not restrictive for our method and can be changed in other realistic situations.

In Case 1, we consider 100 scenarios of surgery durations randomly generated based on the Log-normal distribution of the surgery duration presented in [17]. For Case 2, we assume that a surgery duration follows a normal distribution with the mean and standard deviation reported in [17].

The test problems are categorized in three groups “small”, “medium” and “large” and labelled by the number of the surgeries followed by the number of surgeons e.g.,  $s15 - g7$  consists of 15 surgeries and 7 surgeons. We generated 10 test instances for each size of problems.

When the random surgery durations are modelled by a finite set of discrete approximations (i.e., scenarios)  $\mathcal{S}$  where the probability of realizing scenario  $\zeta$  is  $p_\zeta$ , then the chance constraint (9) is written by  $\Pr(\sum_{\zeta \in \mathcal{S}} p_\zeta \mathbb{I}_{I_\zeta}) \geq 1 - \epsilon$  where  $\mathbb{I}_{I_\zeta}$  is an indicator function and is equal 1 if for scenario  $\zeta$ ,

$$\sum_{i \in I_\zeta} d_i^\zeta + \sum_{i, j \in I_\zeta} \tau_{ij} + \bar{\tau}_{i^*} \leq L.$$

For cases 2 and 3, we assume that the durations follow normal distributions i.e.,  $\bar{d}_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$  where  $\mu_i$  and  $\sigma_i^2$  are the mean and the variance of the duration of surgery  $i$ . The corresponding chance constraint is then reformulated by

$$\Pr \left( \frac{\sum_{i \in I_s} (\tilde{d}_i - \mu_i)}{\sum_{i \in I_s} \sigma_i} \leq \frac{\bar{L} - \sum_{i \in I_s} \mu_i}{\sum_{i \in I_s} \sigma_i} \right) = \Pr \left( \mathcal{Z} \leq \frac{\bar{L} - \sum_{i \in I_s} \mu_i}{\sum_{i \in I_s} \sigma_i} \right) \geq 1 - \epsilon$$

where  $\bar{L} = L - \sum_{i,j \in I_s} \tau_{ij} - \bar{\tau}_i^*$  and  $\mathcal{Z}$  is the standard normal distribution random variable.

### 5.2 Numerical results

In this section, we compare in total nine performance measures categorized into two types: optimization and simulation related measures for our proposed branch-and-price method denoted by proposed method, and the standard formulation of CCP presented by (10) denoted by CPLEX. The optimization related measures investigate the efficiency of the proposed framework and the quality of the optimal solution. Here, we consider six measures in this category as follows: (1) solution time, (2) the integrality gap after 7200 s, (3) the number of variables (sequences) generated during the solution procedure, (4) the number of ORs opened, (5) the objective function value denoted ( $z^*$ ) and (6) the surgeon waiting time denoted by ‘‘Srgn. wait’’ (minutes). The surgeon waiting time reports the expected time a surgeon must wait between two surgeries of his/her surgery list in a time period.

In the second category, we carry out a Monte Carlo Simulation to evaluate the obtained optimal solution of an instance. To this end, 10,000 scenario realizations based on the surgery duration distribution function are generated. Then for the solution obtained by solving the model and for each scenario, the following performance measures are computed. Finally, the average values of these measures are calculated and reported in the tables. The performance measures are (1) the average reliability of the solution for exceeding the session length ( $\frac{\sum_{r=1}^{m'} (\sum_{s \in S} p_s |\tilde{D}_r > L|)}{m'}$ ), denoted by ‘‘Prob. Exceed.’’, (2) the average expected time of exceeding session length for open ORs i.e.,  $\frac{\sum_{r=1}^{m'} \mathbb{E}[\tilde{D}_r - L]^+}{m'}$  in minutes, denoted by ‘‘Overtime’’, and (3) the average of the expected time of unused session length i.e.,  $\frac{\sum_{r=1}^{m'} \mathbb{E}[L - \tilde{D}_r]^+}{m'}$  in minutes, denoted by ‘‘Unus. Cap.’’ where  $m'$  is the total number of final sequences,<sup>1</sup>  $[a]^+ = \max(a, 0)$  and  $\tilde{D}_r$  is the total duration of surgeries assigned to OR  $r$ .

We implement the set-partitioning based models in SCIP (Solving Constraint Integer Programming) software. To demonstrate the computational efficiency of the proposed method, we implement the standard CCP model is in ILOG IBM CPLEX 12.51 since this software is commonly used and also is known to be a very powerful branch-and-cut solver. The models are run on a OSX with 3.1 GHz Intel Core i5 and 8 GB of memory. We give another advantage to the standard CCP model by letting ILOG IBM CPLEX to use its parallel computing feature, while SCIP can only use one processor during solving the instances. Finally, we set 7200-s limitation of the solution time.

<sup>1</sup> If the optimality gap is zero, then final sequences refers to the optimal ones used to assign surgeries.

### 5.2.1 Efficiency of the method

In order to demonstrate the efficiency of the proposed approach, we compare our results firstly with those of related works in the literature, and secondly with those obtained from solving the same instances by the standard chance constraint model using ILOG IBM CPLEX. The latter also demonstrates the correctness of our approach.

Neyshabouri and Berg [19] use the same data set i.e., [17]. They applied robust optimization to their problem and were able to solve up to 40 surgeries. Although they consider downstream activities, they do not take into account surgeons and turnover time. In another related paper, Batun et al. [4] formulated their problem by a two-stage stochastic integer model. The largest instances that they were able to solve, consist of 11 surgeries, 3 surgeons, and 6 ORs in a surgical day. A set of 500 scenarios is used to represent the uncertainty. Using a progressive hedging algorithm, Gul et al. [12] found near optimal solutions for instances up to 50 surgeries for a planning period of 10 days. They do not consider surgeons in their problem, but they include cancellation cost.

We solve instances of Case 1 using the proposed method by SCIP and the standard CCP model by ILOG IBM CPLEX 1251. ILOG IBM CPLEX solved  $s_4 - g_4$  with 100 scenarios in 7.10s, but it was unable to solve or find an integer solution for any other instances. These cases are indicated by “NS” (Not Solved) in Table 1. This table and Table 2 report the results of Case 1 and the standard model. As the results present, the proposed method was able to solve all instances to optimality except the last one for which the integrality gap was 1 percent. Table 2 provides more details of Case 1 including solution time and integrality gap. Linear programming (LP) relaxation of the instances is solved to find a lower bound and the LP relaxation gap for the standard model. We use the optimal solution obtained by the proposed method to compute the LP relaxation gap. As expected, LP relaxation provides a poor lower bound for this class of problem. The solution time for the LP relaxation of the instances reported are considerably high such that instance  $s_{51} - g_{13}$  was solved in over 5 h and the model for the last instance ( $s_{60} - g_{13}$ ) was not even constructed after 2 h of time limit so the lower bound for this case was not available (indicated by “NA”). In contrast with the standard model which generated a large number of binary variables in the beginning, the proposed method was able to solve the instances to optimality with much fewer variables. The number of variables for the standard model presented in the table, is the number of variables in the reduced problem provided by ILOG IBM CPLEX pre-solution procedure which reduces the problem size prior to passing to its optimizer. As the results suggests, our proposed method was significantly faster to solve the instance and lead to the same solution and the same objective function value of the standard model.

Table 2 reports the average results of solving Cases 1 and 2 for the following performance measures: the solution time, the integrality gap (percent) after 7200s and the number of variables added to the master problem. As the table shows, the proposed method was very quick to solve small and medium instances. The solutions obtained for the large instances were very close to the optimality with maximum 2% gap on average for the test instances of  $s_{60} - g_{13}$ . The solution time and the number of sequences (variables) added to the problems were considerably higher in Case 1 in

**Table 1** The average performance measures for verifying the correctness of the proposed method against the standard CCP model

Type	Instance	Proposed method-Case 1		CPLEX-Case 1			LP sol. time (s)	LP gap (%)
		Optimum $z^*$	# variables	Optimum $z^*$	# variables	$z_{LP}$		
Small	s4 - g4	12,717	15	12,717	5494	12,498	0.29	1.72
	s10 - g3	25,455	99	NS	7645	18,725	39.94	26.44
	s15 - g7	38,184	263	NS	10,690	18,724	79.52	50.96
Medium	s20 - g7	44,531	2128	NS	16,282	24,963	219.25	43.94
	s25 - g8	44,961	894	NS	18,777	24,965	505.43	44.47
	s31 - g8	70,400	6914	NS	39,690	18,722	2962.25	73.41
Large	s40 - g10	89,990	3720	NS	84,586	18,723	19,122.20	79.19
	s51 - g13	108,430	7471	NS	164,377	NS	NA	NA
	s60 - g13	102,238	7521	Problem was not constructed in 2h				

**Table 2** The average performance measures of the proposed method for 10 test problems of each instance size for Cases 1 and 2

Type	Instance	Case 1			Case 2		
		Sol. time (s)	Int. gap (%)	# added variables	Sol. time (s)	Int. gap (%)	# added variables
Small	s4 – g4	0.00	0.00	5.20	0.00	0.00	5.30
	s10 – g3	0.03	0.00	55.60	0.04	0.00	50.40
	s15 – g7	0.12	0.00	117.38	0.14	0.00	105.40
Medium	s20 – g7	0.79	0.00	303.70	0.62	0.00	248.67
	s25 – g8	1.04	0.00	345.60	0.58	0.00	282.89
	s31 – g8	156.81	0.00	1818.30	68.30	0.00	1161.50
Large	s40 – g10	893.57	0.00	3527.00	254.10	0.00	1923.50
	s51 – g13	4962.85	1.23	6502.70	467.17	0.00	3011.40
	s60 – g13	6016.16	2.07	7841.90	471.15	0.00	3567.50

comparison with Case 2. The reason is that the continuous approximation is more conservative and a sequence which is feasible under the assumption of Case 1 may not be feasible under the assumption of Case 2. Therefore, the search space is more limited in Case 2. In the next sections, in order to be able to compare the results of cases, the result of only one test problem is reported for each instance.

### 5.2.2 Case 1 versus case 2

It is common to use the discrete scenario approximation of uncertain parameters in stochastic programming, however, its accuracy highly depends on the size of scenario set which usually affects the solution quality and the computational efficiency. The larger of scenario set, the better solution quality and the poorer computational efficiency. In Table 3, we investigate the quality of the solution for Case 1 and case 2. While the number of the ORs opened and the objective function value are higher in Case 2, the expected probability of failure obtained by Monte Carlo simulation, is considerably lower in Case 1. Column “Prob. Exceed” and also the “Average” row show that Case 1 does not hold the probabilistic constraint of the session length i.e., the solutions do not meet constraint (9) requirements. As a result, the expected exceeding time and its average are larger in Case 1. Smaller unused capacity is preferred only when constraint (9) is held, which is not the case in Case 1. The conclusion is that when we can incorporate the distribution function or a relatively good continuous approximation for the chance constraints, there may be no gain in terms of the solution quality and the computational efficiency to use the discrete approximation of the original distribution function specially since the scenario generation itself is a complicated task. Furthermore as the integrality gap suggests, the continuous approximation of the surgery duration did not increase the complexity of the resulting problem and all instances were solved to optimality in the time limit.



**Table 3** The comparison of the performance measures for Case 1 and Case 2

Instance	Case 1				Case 2					
	Optimization		Simulation		Optimization		Simulation			
	#ORs opened	Int. gap (%)	Prob. Exceed.	Overtime (min)	Unus. Cap (min)	#ORs opened	Int. gap (%)	Prob. Exceed.	Overtime (min)	Unus. Cap. (min)
s4 – g4	2	0	0.1	3.3	193.6	2	0	0.05	1.8	191.5
s10 – g3	3	0	0.3	19.7	53.1	4	0	0.07	2.6	129.6
s15 – g7	5	0	0.2	9.3	82.8	6	0	0.09	3.6	133.3
s20 – g7	6	0	0.3	14.9	71.6	7	0	0.11	4.2	112.5
s25 – g8	7	0	0.5	40.2	29.4	10	0	0.07	2.5	112.8
s31 – g8	9	0	0.4	30.3	41.3	11	0	0.15	6.2	82.1
s40 – g10	12	0	0.3	19.8	62.8	15	0	0.09	3.2	118
s51 – g13	15	6	0.3	18.7	54	18	0	0.12	5.1	97.7
s60 – g13	16	0	0.5	31.6	35.3	20	0	0.13	5.5	92.2
Average			0.33	20.88	69.33			0.097	3.85	118.88

### 5.2.3 Case 2 versus case 3

Here, we compare two types of the objective functions. As mentioned earlier, Case 3 is similar to Case 2 except in its objective function in which only ORs' opening cost is minimized i.e.,  $\min \sum_{t \in T} \sum_{s \in S_t} c_s u_s$  where  $c_s = c_R$ . Including turnover penalty in the objective function has two consequences. First, it is highly possible to have multiple optimal solutions in Case 3 that results in opening the same number of ORs with different assignments. In the presence of the turnover penalty, those solutions with the minimum total duration is chosen. As reported in Table 4, the number of ORs opened in all the instances for the both models are the same while the overtime and the unused capacity of Case 2 are in all the cases smaller (except two cases) and bigger than those of Case 3, respectively.

Secondly, a proper choice of turnover penalty could reduce the surgeons' waiting time or could result in a desired arrangement for surgeons. As mentioned in the data set explanation, the turnover time between two surgeries is less if the two surgeries are performed by the same surgeon. Therefore, by assigning higher penalty to turnover times in the objective function, the model tends to put all the surgeries assigned to a surgeon in a row if it is feasible to do so. As a result, the surgeon waiting time is minimized indirectly. This is achieved without changing our model or increasing the computational complexity. Figure 1 illustrates an example for two sequences of the optimal solution for Instance s51 – g13 based on Case 3. In this figure, the empty red boxes present the turnover time and the numbers in and above the blue boxes indicates the surgeries' and the surgeons' codes, respectively. As can be seen, there is a gap between two surgeries of surgeon 5 and surgeon 7. Therefore, they will have to wait to perform their next surgeries. Considering positive turnover penalty cost can avoid the waiting time by rearranging the surgeries while the total number of in-service ORs are kept at the same level. This issue has been mainly neglected in the literature.

Column "Srgn. waiting" in Table 4 reports the surgeon waiting time and the average for all the instances. In Case 3 surgeons have to wait up to 180min while with the same number of in-service ORs, the surgeon's waiting time is zero in Case 2.

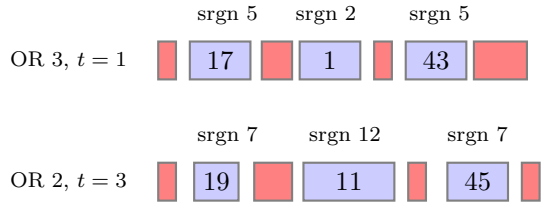
## 5.3 Sensitivity analysis

Choosing input parameter  $\epsilon$  has a significant impact on the decision variables and the objective function value. In addition to OR opening cost and turnover penalties, there exist two other costs associated with violating session lengths and unused capacity in ORs opened. Due to randomness of the surgery durations, the OR session length is either violated or left unused for each value of  $\epsilon$ . The both situations result in costs and may not be desirable for the hospital management. The amount of overtime is in conflict with the unused capacity in opened ORs. The hospital management would like to choose a value for  $\epsilon$  in which the total cost consisting of the objective function and the costs associated with overtime and unused capacity, are minimum. This trade-off has been mainly neglected in the literature. Including these two in the formulation phase makes the problem more difficult to model and solve. Although in theory this integration can be done within the two-stage chance constrained programming frame-

**Table 4** The comparison of the performance measures for Case 2 and Case3

Instance	Case 2				Case 3					
	Optimization		Simulation		Optimization		Simulation			
	#ORs opened	Int. gap (%)	Sign wait (min)	Overtime (min)	Unused Cap. (min)	#ORs opened	Int. gap (%)	Sign. wait (min)	Overtime (min)	Unused Cap. (min)
s4 - g4	2	0	0.0	1.8	191.5	2	0	0.0	0	182.9
s10 - g3	4	0	0.0	2.6	129.6	4	0	0.0	2.6	127.4
s15 - g7	6	0	0.0	3.6	133.3	6	0	0.0	2.4	125.8
s20 - g7	7	0	0.0	4.2	112.5	7	0	0.0	5.1	97.9
s25 - g8	10	0	0.0	2.5	112.8	10	0	0.0	3.2	98.9
s31 - g8	11	0	0.0	6.2	82.1	11	0	162.9	6.7	79.4
s40 - g10	15	0	0.0	3.2	118	15	0	124.2	3.9	103.4
s51 - g13	18	0	0.0	5.1	97.7	18	0	144.0	5.3	90.9
s60 - g13	20	0	0.0	5.5	92.2	20	0	180.0	6.3	83.8
Average			0.00	3.85	118.88			67.9	3.94	110.06

**Fig. 1** Schedules of instance s51 – g13 illustrating the gaps between surgeries of a surgeon in the optimal solution of Case 3



work, only very small instances under a limited setting can be solved [13]. Instead, the sensitivity analysis for  $\epsilon$  provides us with useful information to choose the right value of  $\epsilon$  for which the total cost is minimized.

In this section, we study the impact of  $\epsilon$  on the total cost defined above. We focus on the sensitivity of four measures including the number of OR opened, the objective function and the average of the expected values of the overtime and the unused capacity, with respect to the variation of  $\epsilon$ . In addition to  $\epsilon = 0.10$ , six other values for  $\epsilon$  i.e.,  $\epsilon \in \{0.01, 0.05, 0.15, 0.20, 0.25, 0.30\}$  are chosen.

Due to lower semi-continuity of the objective function with respect to  $\epsilon$ , an increase (or decrease) in  $\epsilon$  does not necessarily lead to an increase (or decrease) in the objective function and the number of the ORs opened. As reported in Table 5, the results show that the number of ORs opened is less sensitive to the variations in  $\epsilon$  than the objective function value. This observation suggests that in order to achieve a more reliable solution, we do not always need to increase the number of in-service ORs but it can be done by improving the assignment of surgeries to the ORs.

The chance constraint (9) is directly connected to the overtime; therefore although not continuously, the overtime increases by  $\epsilon$ . This trend can be seen in our results, see Column “Overtime” of Table 5. An opposite behaviour is expected for the unused capacity. However, our results shows that the behaviour of the unused capacity cannot be linearly predicted since it has not been explicitly considered in the problem.

We further extend our experiment for s60 – g13 and solve the instance for  $\epsilon = 0.35, 0.40, 0.45$  and  $0.50$ . We compute four measures: the mean, standard deviation, 95% quantile and worst case values of the simulation experiment for the overtime and unused capacity. The results are reported in Table 6. The objective function and the number of ORs opened decrease as  $\epsilon$  increases. Figure 2 illustrates the trend of changing the distribution for the overtime and unused capacity when  $\epsilon$  changes. The upper bound of the interval is 95% quantile and its lower bound is  $\bar{x} - (q - \bar{x})$  if it is not negative and otherwise zero, where  $\bar{x}$  and  $q$  are the mean and 95% quantile of overtime or unused capacity. Note that in the figure the cost functions are not continuous as presented. For the sake of presentation, the segments between the points are drawn.

The results suggests that although very small values of  $\epsilon$  result in very small overtime, they lead to very high unused capacity and impose unnecessary costs. Given the above information and the hospital’s strategy, hospital managers can choose a right value for  $\epsilon$  to male a reasonable trade-off between their operation cost.

**Table 5** The sensitivity of the solution to variations in  $\epsilon$

Instance	$\epsilon$	#ORs opened	$z^*$	Overtime (min)	Unused Cap. (min)	Instance	$\epsilon$	#ORs opened	$z^*$	Overtime (min)	Unused Cap. (min)
s4 – g4	0.01	2	12,741	0	183.43	s31 – g8	0.01	14	89,359	0.73	152.98
	0.05	2	12,733	1.9	189.01		0.05	12	76,742	3.2	102.18
	0.10	2	12,728	1.8	191.54		0.10	12	76,712	3.81	105.53
	0.15	2	12,727	3.4	193.86		0.15	11	70,424	7.51	79.28
	0.20	2	12,727	3.47	193.72		0.20	11	70,374	6.99	83.47
s10 – g3	0.25	2	12,727	3.54	193.31	s40 – g10	0.25	11	70,320	10.28	91.68
	0.30	2	12,727	3.4	194.06		0.30	10	64,030	14.04	60.77
	0.01	5	31,760	0.47	185.16		0.01	17	108,299	0.81	146.02
	0.05	4	25,457	2.03	128.91		0.05	15	95,640	2.64	115.04
	0.10	4	25,456	2.56	129.64		0.10	15	95,605	3.19	118.01
s15 – g7	0.15	4	25,442	3.72	134.34	s51 – g13	0.15	14	89,279	4.98	99.82
	0.20	4	25,442	3.75	134.15		0.20	13	83,003	8.67	76.85
	0.25	4	25,442	3.88	134.1		0.25	13	82,961	10.43	81.79
	0.30	3	19,144	19.62	52.76		0.30	12	76,704	16.67	54.82
	0.01	7	44,504	0.5	169.73		0.01	21	133,805	1.04	135.7
	0.05	6	38,204	1.88	128.88		0.05	19	121,114	2.32	111.8
	0.10	6	38,187	3.61	133.34		0.10	18	114,808	5.11	97.74
	0.15	6	38,183	4.13	134.25		0.15	17	108,425	7.09	85.37
	0.20	6	38,183	4.05	134.54		0.20	17	108,406	8.41	87.87
	0.25	5	31,878	9.31	82.62		0.25	16	102,096	11.59	69.85
0.30	5	31,878	9.25	83.39	0.30	15	95,815	18.49	51.72		

**Table 5** continued

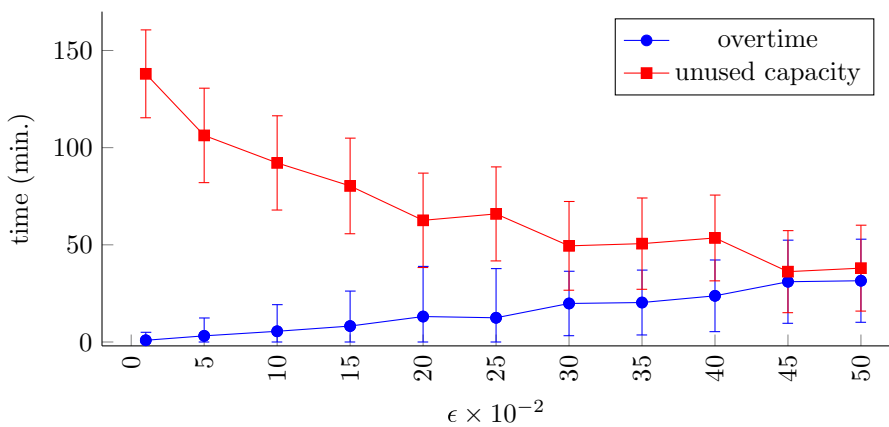
Instance	$\epsilon$	#ORs opened	$z^*$	Overtime (min)	Unused Cap. (min)	Instance	$\epsilon$	#ORs opened	$z^*$	Overtime (min)	Unused Cap. (min)
s20 – g7	0.01	8	50,870	0.76	143.7	s60 – g13	0.01	24	152,767	0.89	137.99
	0.05	7	44,537	2.58	110.61		0.05	21	133,738	3.16	106.3
	0.10	7	44,533	4.24	112.55		0.10	20	127,445	5.51	92.15
	0.15	7	44,531	5.8	114.69		0.15	19	121,092	8.18	80.30
	0.20	6	38,246	12	66.83		0.20	18	114,840	13.10	62.61
	0.25	6	38,237	12.43	68.49		0.25	18	114,780 <sup>♣</sup>	12.46	65.91
	0.30	6	38,233	13.13	69.92		0.30	17	108,518 <sup>♣</sup>	19.82	49.47
s25 – g8	0.01	12	76,563	0.81	160.99						
	0.05	10	63,933	2.33	113.39						
	0.10	10	63,933	2.32	113.36						
	0.15	9	57,618	6.14	84.65						
	0.20	9	57,603	6.36	86.43						
	0.25	9	57,584	7.93	90.06						
	0.30	9	57,581	7.97	90.47						

♣ indicates that the integrality gap is not zero for those instances. The maximum of the integrality gap is 3.73%. Overtime and Unused Cap. times are given per opened OR per day

**Table 6** The sensitivity analysis for Instance s60 – g13

Overtime/unused Cap.	$\epsilon$	# ORs opened	$z^*$	Int. gap (%)	Mean (min)	Std. (min)	95% quantile (min)	Worst (min)
Overtime	0.01	24	152,767	0.00	0.89	1.5	4.1	8.5
Unused Cap.					137.99	13.5	160.6	185.6
Overtime	0.05	21	133,738	0.00	3.16	3.1	9.2	24.9
Unused Cap.					106.3	14.6	130.6	157.5
Overtime	0.10	20	127,445	0.00	5.51	4.3	13.7	29.2
Unused Cap.					92.15	14.5	116.4	149.9
Overtime	0.15	19	121,092	0.00	8.20	5.4	18.0	34.9
Unused Cap.					80.30	14.8	104.9	140.6
Overtime	0.20	18	114,840	0.00	13.10	7.2	25.8	47.5
Unused Cap.					62.61	14.1	86.9	118.6
Overtime	0.25	18	114,780	3.73	12.46	6.9	25.3	45.1
Unused Cap.					65.91	14.3	90.1	120.7
Overtime	0.30	17	108,518	0.16	19.82	9.2	36.4	61.4
Unused Cap.					49.47	13.5	72.3	103.1
Overtime	0.35	17	108,507	3.49	20.31	9.3	37.0	67.1
Unused Cap.					50.61	13.6	74.1	121.0
Overtime	0.40	17	108,516	5.61	23.76	10.2	42.2	75.9
Unused Cap.					53.53	12.9	75.6	106.6
Overtime	0.45	16	102,215	1.33	31.01	11.9	52.4	93.9
Unused Cap.					36.19	12.1	57.3	87.4
Overtime	0.50	16	102,197	3.49	31.52	12.2	52.9	85.4
Unused Cap.					38.00	12.4	60.1	90.9

Overtime and unused Cap. times are given per open ORs per day



**Fig. 2** Sensitivity analysis for Instance s60 – g13. The intervals show the 95 quantile and illustrate the trend of changing the distribution for the overtime and unused capacity when  $\epsilon$  changes

## 6 Conclusion

We have studied an integrated surgery scheduling and planning problem with uncertain surgery durations. The uncertainty in OR capacity constraints is treated by chance constrained programming. The resulting problem is challenging due to its structure which is a combination of stochastic optimization and integer programming. We propose a set-partitioning formulation for the underlying problem. The associated column generation subproblem is decomposed over ORs and time periods. The subproblem is then reformulated as a shortest path problem for which a search algorithm has been devised using the optimality and feasibility conditions. This reformulation allows incorporating distribution functions as well as scenario representation of surgery durations into the chance constraints while keeping the complexity of the problem tractable. A significant reduction in solution time has been achieved due to efficiently solving the resulting shortest path problem. Computational results show that the proposed method is able to solve larger instances of the problem than those previously reported for a similar setting.

An extensive simulation study has been performed to assess the quality of optimal solutions. Several performance measures including the surgeons' waiting time, expected probability of exceeding OR session length, expected overtime and unused capacity have been evaluated. In comparison with discrete scenarios, working with a continuous approximation of distribution functions for surgery durations lead to reduction of the expected probability of exceeding OR session length and overtime. However, the unused capacity would be increased. Our computational results suggest that adding a proper turnover penalty cost to the objective function would reduce the surgeons' waiting time while the number of in-service ORs, overtime and unused capacity would change slightly.

A sensitive analysis has been carried out to investigate the impact of the probability of exceeding OR session length on the number of opened ORs, OR overtime and unused capacity. We conclude that a high reliability level for holding OR session length constraint is not always the best choice as it may impose unnecessary costs.

The performance of the proposed method highly depends on the search space of the column generation subproblem and efficiency of the employed branch-and-bound method. The size of search space increases with the number of surgeons, number of planning horizon and number of surgeries. We used a modified dynamic programming search algorithm to solve the reformulated subproblem. The efficiency of the proposed method could be improved by running faster algorithms such as meta heuristics.

To extend this work, one could solve a more practical problem by taking OR downstream activities into account. The proposed method may also be applied to other situations where probabilistic bin packing constraints exist.

## Appendix: Standard chance constrained model for the stochastic surgery scheduling problem

Here, we present the standard chance constrained equivalent model using the popular scenario based chance constrained programming [14]. The notation for column gen-



eration (6) are used to define the standard model. The additional binary variables  $z_r^t$  and  $q_k^{r,t}$  indicate if OR  $r$  is used in period  $t$ , and if surgeon  $k$  is allocated to OR  $r$  in period  $t$ .

$$\min \sum_{r \in R} \sum_{t \in T} c_R z_r^t + \sum_{i \in S} \sum_{j \in I} \sum_{l \in I} c_{ij} \tau_{ij} y_{ij}^l + \sum_{i \in I} \sum_{t \in T_i} \sum_{r \in R} c_i \bar{\tau}_i w_i^{r,t} \tag{10a}$$

$$\text{s.t.: } \sum_{l \in I} \sum_{r \in R} \sum_{t \in T_i} x_{i,l}^{r,t} = 1, \forall i \in I \tag{10b}$$

$$\sum_{l \in I} x_{i,l}^{r,t} \leq z_r^t, \forall i \in I, r \in R, t \in T \tag{10c}$$

$$\sum_{i \in I_k} \sum_{l \in I} x_{i,l}^{r,t} \leq M q_k^{r,t}, \forall k \in K, r \in R, t \in T \tag{10d}$$

$$\sum_{r \in R} q_k^{r,t} \leq 1, \forall k \in K, t \in T \tag{10e}$$

$$\sum_{i \in I} x_{i,l}^{r,t} \geq \sum_{i \in I} x_{i,l+1}^{r,t}, \forall l \in I, r \in R, t \in T \tag{10f}$$

$$x_{i,l}^{r,t} \leq \sum_{j \in I} x_{j,l+1}^{r,t} + w_i^{r,t}, \forall i, l \in I, r \in R, t \in T \tag{10g}$$

$$w_i^{r,t} \leq \sum_{l \in I} x_{i,l}^{r,t}, \forall i \in I, r \in R, t \in T \tag{10h}$$

$$\sum_{t \in T} x_{i,l}^{r,t} + \sum_{t \in T} x_{j,l+1}^{r,t} - 1 \leq y_{ij}^l, \forall i, l \in I, r \in R, t \in T \tag{10i}$$

$$y_{ii} \geq x_{i0}^{r,t} \forall i \in I, r \in R, t \in T \tag{10j}$$

$$\sum_{i,l \in I} d_i^\zeta x_{i,l}^{r,t} + \sum_{i,j \in I} \tau_{ij} y_{ij}^l + \sum_{i \in I} \bar{\tau}_i \delta_i^{r,t} \leq L + M(1 - \pi_\zeta^{r,t}), \forall r \in R, t \in T, \zeta \in S \tag{10k}$$

$$\sum_{\zeta \in S} p_\zeta \pi_\zeta^{r,t} \geq 1 - \alpha, \forall r \in R, t \in T \tag{10l}$$

$$\sum_{r=1}^{\min\{k, |R|\}} q_k^{r,t} \leq \sum_{r=1}^{\min\{k, |R|\}} z_r^t, \forall k \in K, t \in T \tag{10m}$$

$$x_{i,l}^{r,t}, y_{ij}^l, z_r^t, w_i^{r,t}, q_k^{r,t}, \pi_\zeta^{r,t} : \text{Binary } \forall i, l, j \in I, \zeta \in S, r \in R, t \in T. \tag{10n}$$

The objective function computes the opening cost and the penalty cost. Constraint (10b) corresponds with the first constraint of SP ensuring every surgery is assigned to a OR at some period. Note that  $v_{it} = \sum_{l \in I} \sum_{r \in R} x_{i,l}^{r,t}$ . Constraint (10c) enforces the limitation of the available ORs at each period i.e.,  $z_r^t$  to take a value of 1 if at least one surgery is assigned to OR  $r$  in period  $t$ . This constraint is associated with constraint (1e). Constraints (10d and 10e) are related to constraint (6e) forbidding the assignment of a surgeon to more than one OR in each period. Constraints (10f–10j) determine the arrangement of each sequence. In SP, this is presented by variable  $u_{s_t}$  through the column generation problem. Finally, constraint (10k–10l) impose the probabilistic condition on the OR available length via a scenario-based chance constraint. These constraints are equivalents of the column generation feasibility condition (9). Constraint (10m) is a symmetry breaking constraint and implies that surgeon 1 is assigned

to OR 1, surgeon 2 can be assigned to OR 1 and OR 2 and so on. Finally, Constraints (10n) enforce the integrality conditions on the decision variables.

## References

- Ahmed, S., Papageorgiou, D.J.: Probabilistic set covering with correlations. *Oper. Res.* **61**, 438–452 (2013)
- Baldacci, R., Mingozzi, A.: A unified exact method for solving different classes of vehicle routing problems. *Math. Program.* **120**(2), 347–380 (2009)
- Barnhart, C., Johnson, E., Nemhauser, G., Savelsbergh, M., Vance, P.: Branch-and-price: column generation for solving huge integer programs. *Oper. Res.* **46**, 316–329 (1998)
- Batun, S., Denton, B.T., Huschka, T.R., Schaefer, A.J.: Operating room pooling and parallel surgery processing under uncertainty. *INFORMS J. Comput.* **23**, 220–237 (2011)
- Cardoen, B., Demeulemeester, E., Beliën, J.: Operating room planning and scheduling: a literature review. *Eur. J. Oper. Res.* **201**, 921–932 (2010)
- Charnes, A., Cooper, W.: Chance-constrained programming. *Manag. Sci.* **6**, 73–79 (1959)
- Deng, Y., Shen, S., Denton, B.: Chance-constrained surgery planning under uncertain or ambiguous surgery durations (2015, unpublished). <https://papers.ssrn.com/sol3/papers.cfm?abstractid=2432375>
- Denton, B.T., Miller, A.J., Balasubramanian, H.J., Huschka, T.R.: Optimal allocation of surgery blocks to operating rooms under uncertainty. *Oper. Res.* **58**, 802–816 (2010)
- Erdogan, S.A., Denton, B.T.: *Wiley Encyclopedia of Operations Research and Management Science*. Wiley, New York (2011). ch. Surgery planning and scheduling
- Etzioni, D., Liu, J., Maggard, M., Ko, C.: The aging population and its impact on the surgery workforce. *Ann. Surg.* **238**, 170–177 (2003)
- Fukasawa, R., Longo, H., Lysgaard, J., Aragão, M.P.D., Reis, M., Uchoa, E., Werneck, R.F.: Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Math. Program.* **106**, 491–511 (2006)
- Gul, S., Denton, B.T., Fowler, J.W.: A progressive hedging approach for surgery planning under uncertainty. *INFORMS J. Comput.* **27**, 755–772 (2015)
- Liu, X., Kucukyavuz, S., Luedtke, J.: Decomposition algorithms for two-stage chance-constrained programs. *Math. Program. Ser. B* **157**, 219–243 (2014)
- Luedtke, J.: A branch-and-cut decomposition algorithm for solving chance-constrained mathematical programs with finite support. *Math. Program.* **146**, 1–26 (2013)
- Lulli, G., Sen, S.: A branch-and-price algorithm for multistage stochastic integer programming with application to stochastic batch-sizing problems. *Manag. Sci.* **50**, 786–796 (2004)
- Meng, F., Qi, J., Zhang, M., Ang, J., Chu, S., Sim, M.: A robust optimization model for managing elective admission in a public hospital. *Oper. Res.* **63**, 1452–1467 (2015)
- Min, D., Yih, Y.: Scheduling elective surgery under uncertainty and downstream capacity constraints. *Eur. J. Oper. Res.* **206**, 642–652 (2010)
- Nemirovski, A., Shapiro, A.: Convex approximations of chance constrained programs. *SIAM J. Optim.* **17**, 969–996 (2006)
- Neyshabouri, S., Berg, B.: Adaptive elective surgery planning under duration and length-of-stay uncertainty: a robust optimization approach. (2015)
- Noorizadegan, M.: On vehicle routing with uncertain demands. Ph.D. thesis, Warwick Business School (2013)
- Pessoa, A., de Aragao, M.P., Uchoa, E.: A robust branch-cut-and-price algorithm for the heterogeneous fleet vehicle routing problem. *Lect. Notes Comput. Sci.* **4525**, 150–160 (2007)
- Pulido, R., Aguirre, A.M., Ortega-Mier, M., García-Sánchez, A., Méndez, C.A.: Managing daily surgery schedules in a teaching hospital: a mixed-integer optimization approach. *BMC Health Serv. Res.* **14**, 1–13 (2014)
- Saxena, A., Goyal, V., Lejeune, M.A.: Mip reformulations of the probabilistic set covering problem. *Math. Program.* **121**, 1–31 (2010)
- Sherali, H.D., Zhu, X.: Two-stage stochastic mixed-integer programs: algorithms and insights. In: Gao, D.Y., Sherali, H.D. (eds.) *Advances in Applied Mathematics and Global Optimization*, pp. 405–435. Springer Science+Business Media, Boston, MA (2009)

25. Shylo, O.V., Prokopyev, O.A., Schaefer, A.J.: Stochastic operating room scheduling for high-volume specialties under block booking. *INFORMS J. Comput.* **25**, 682–692 (2013)
26. Song, Y., Luedtke, J.R., Küçükyavuz, S.: Chance-constrained binary packing problems. *INFORMS J. Comput.* **26**, 735–747 (2014)
27. Wang, Z., Crowcroft, J.: Quality-of-service routing for supporting multimedia applications. *IEEE Sel. Areas Commun.* **14**, 1228–1234 (1996)
28. Wang, Y., Tang, J., Fung, R.Y.K.: A column-generation-based heuristic algorithm for solving operating theater planning problem under stochastic demand and surgery cancellation risk. *Int. J. Prod. Econ.* **158**, 28–36 (2014)
29. Zhang, B., Murali, P., Dessouky, M.M., Belson, D.: A mixed integer programming approach for allocating operating room capacity. *J. Oper. Res. Soc.* **60**, 663–673 (2009)
30. Zhang, Z., Xie, X.: Simulation-based optimization for surgery appointment scheduling of multiple operating rooms. *IIE Trans.* **47**, 998–1012 (2015)