Stochastic patient admission scheduling with an exponential number of scenarios

Haichao Liu [a,b,c], Yang Wang [a] (Corresponding author), Jin-Kao Hao [b] and Abraham P. Punnen [c]

[a] School of Management, Northwestern Polytechnical University, 710072 Xi'an, China

[b] LERIA, Université d'Angers, 49045 Angers, France

[c] Department of Mathematics, Simon Fraser University, Surrey, V3T 0A3, Canada

IISE Transactions, Aug. 2026

Abstract

Overstay patients—those who remain hospitalized longer than planned—generate substantial operational costs, worsen bed shortages, and delay care for other patients. This paper studies a stochastic variant of the well-known patient admission scheduling problem, which aims to assign patients to rooms during hospitalization while considering uncertain overstay lengths. We build two-stage stochastic programming models to formulate the problem, where the first stage assigns patients to rooms during their fixed length of stay, and the second stage evaluates the expected costs resulting from patient overstay. Compared to the typical scenario-based model, the proposed state-variable (SV) model is significantly reduced by having a pseudo-polynomial number of variables and constraints. To solve the SV model efficiently, we introduce the sample average approximation (SAA) method as the first attempt to provide a high-quality initial feasible solution for the Gurobi solver. To support practical decision-making under dynamic patient arrivals, the SAA-SV method is further embedded into a rolling horizon planning framework. Experimental results show that our SAA-SV method outperforms the methods of directly solving the scenario-based model and the state-variable model in terms of solution quality and computational time. In particular, the SAA-SV method can provide high-quality solutions with an average optimality gap of 3.75% for instances reaching 500 patients with 1-day overstay and 3.27×10^{150} scenarios within 1 hour. Additional analysis has also been carried out to verify the advantage of our proposed method over the typical deterministic approach and the SAA method.

Keywords: Healthcare management; patient admission scheduling; two-stage stochastic programming; state-variable model; rolling horizon planning.

1 Introduction

Hospitals are under pressure from an increase in demand for inpatient care and high-quality patient care. However, it is challenging to meet patients' needs under limited hospital resources (Wang and Demeulemeester, 2023). Among the healthcare services, admitting

patients to the hospital is generally the first step to receive treatment since emergency and surgical care are heavily dependent on bed availability (Ortiga et al., 2012). Limited bed availability can result in long waiting times for patients, which can negatively impact the quality of care and patient satisfaction. To improve the bed utilization, the Patient Admission Scheduling (PAS) problem, which assigns patients to beds in specific rooms on each day of their hospitalization, is of particular significance since its solution impacts many other healthcare-related problems, such as the surgery scheduling problem (Yang et al., 2025) and the nurse scheduling problem (Brandt et al., 2025b).

This paper aims to support healthcare managers in making patient admission decisions by providing methodologies to cope with uncertainty in the inpatient length of stay (LOS). In practice, hospitals predict the LOS for each patient based on their characteristics, physiologic condition, and other factors, to support admission scheduling decisions (Turgeman et al., 2017). However, the LOS is influenced by various real-world factors, such as patients' activities, postoperative recovery, and resource availability (e.g., specialist doctors, operating rooms, and beds). Thus, the actual LOS may be longer than the planned LOS, which is referred to as overstay. Overstay patients can cause bed shortage, backlog for other patients, and further affect the income of the hospital (Chuang et al., 2018). Consequently, overstay patients pose a significant challenge to hospital management.

The dynamic PAS problem, also known as PASU (U for uncertainty), extends the classic PAS problem by considering several real-world features, such as new patient arrivals, delayed admissions, and overstay lengths (Ceschia and Schaerf, 2012). However, most previous studies (Lusby et al., 2016; Guido, 2024) mainly focus on addressing the uncertainty of patient arrivals, and treat overstays by predefining a set of potentially overstay patients with fixed estimated overstay lengths. This deterministic treatment ignores the randomness in overstay lengths and may reduce the robustness of scheduling decisions. In contrast, we investigate the PAS problem under uncertain overstay lengths, referred to as the stochastic PAS (SPAS) problem. The goal is to assign patients to beds over a planning horizon to maximize treatment efficiency, patient satisfaction, and hospital utilization, while satisfying essential medical constraints and accommodating patient preferences as much as possible. A key feature of SPAS is that the LOS for each patient is decomposed into a fixed (planned) LOS and a stochastic overstay length. The fixed LOS refers to the estimated LOS provided

by physicians at the time of admission, based on the patient's condition and treatment plan, while the stochastic overstay length is estimated from historical data and is intended to capture the observed variability in actual LOS among similar patients. This modeling approach not only aligns with clinical practice by scheduling patients based on planned discharge dates while incorporating overstay risks, but also confines the stochastic space to the overstay component alone, leading to a more computationally tractable formulation.

Stochastic programming (SP) (Khatami et al., 2022) is a representative approach for handling healthcare scheduling problems under uncertainty. Scenario-based SP uses a set of scenarios to characterize the LOS and each scenario represents a possible realization. For this approach, the number of scenarios generally grows exponentially with the number of patients because the LOS of all patients are mutually independent. It is quite difficult to solve such model given that each scenario corresponds to specific variables and constraints. Sample average approximation (SAA) (Porras et al., 2023) is a widely-used approach for dealing with a large number of scenarios, which uses a subset of scenarios to avoid creating a large number of variables and constraints. However, SAA cannot guarantee the optimality of the obtained solution. Therefore, dealing with an exponential number of scenarios to obtain the optimal solution is quite challenging when solving the SPAS problem.

The state-variable (SV) modeling approach, proposed by Doulabi et al. (2022), presents a new mixed-integer programming modeling approach for a class of two-stage SP problems, which is capable of modeling problems with an exponential number of scenarios without any need for sampling. This approach has been verified on project scheduling and operating room allocation problems (Doulabi et al., 2022), and operating room planning problem (Hashemi Doulabi and Khalilpourazari, 2023). These studies have demonstrated the effectiveness of the SV approach. Inspired by these studies, our research applies the SV approach to the SPAS problem, presenting a customized formulation and an efficient solution method. The main contributions of our work are summarized as follows.

(1) We study the SPAS problem within a two-stage stochastic programming framework, where the first stage assigns patients to rooms during their fixed LOS, and the second stage evaluates the expected cost resulting from patient overstay.

(2) We propose a scenario-based model $SPAS_{SB}$, which evaluates the expected cost by enumerating all possible scenarios. To overcome its exponential scenario complexity, we

adapt the SV approach to develop the $SPAS_{SV}$ model, reformulating the second-stage problem as multiple independent sequential decision processes with state variables for efficient evaluation. To solve the $SPAS_{SV}$ model efficiently, we propose a new SAA-SV method by using a SAA method to generate an initial feasible solution for the Gurobi solver. To the best of our knowledge, this work is the first successful integration to hybridize SAA method with SV approach to solve SP problems.

(3) Extensive computational experiments demonstrate that our proposed SAA-SV method significantly outperforms the methods of directly solving the $SPAS_{SB}$ model and $SPAS_{SV}$ model. In particular, the SAA-SV method can solve large instances of up to 500 patients with a 1-day overstay and 3.27×10^{150} scenarios within 1 hour, achieving an average optimality gap of 3.75%. In addition, we verify the advantage of the SAA-SV method over the typical deterministic approach and the SAA method.

(4) The SAA-SV method is further embedded into a rolling horizon planning (RHP) framework, named RHP & SAA-SV, and applied to solve real-life cases of the SPAS problem, which considers dynamic patient arrivals. The results show that RHP & SAA-SV is highly effective in managing these dynamic events efficiently.

The remainder of the paper is organized as follows. Section 2 reviews the relevant literature. Section 3 presents the definition of the SPAS problem along with the $SPAS_{SB}$ formulation. Section 4 presents the $SPAS_{SV}$ formulation and the solution method. Section 5 presents the computational results and experimental analysis of the proposed formulations and solution method, along with the integration of the SAA-SV method into an RHP framework and the results of real-life cases. Finally, conclusions and suggestions for future research are given in Section 6.

2 Literature review

This section presents a glimpse of major contribution from the literature closely related to the SPAS problem. Patient admission scheduling is a problem that arises at all planning levels in the healthcare institutes. The strategic level focuses on establishing long-term structure decisions, such as bed configuration (Izady and Mohamed, 2021), that maximize organizational efficiency. The tactical level is concerned with the medium-term planning

decisions to optimize the process organization and execution, such as the formulation of admission policies (Barz and Rajaram, 2015). The operational level focuses on daily admission scheduling and the management of waiting list. At this level, the PAS problem can be divided into deterministic and uncertain versions, thus, the decision-making can be offline or online. The offline situation reflects the in-advance decision making, whereas the online one involves timely response to events that cannot be planned. Additionally, the primary difference between these studies are the uncertainty considered, and the constraints that must be satisfied (hard) or can be violated (soft), and the mathematical model and algorithm used to solve the problem. Since SPAS problem lies at the operational level, we provide a brief overview of the existing PAS research at this level.

The classic operational level PAS problem is a deterministic offline problem. The objective is to assign patients to beds in specific rooms on each day of their hospitalization, while satisfying a set of hard and soft constraints. In the classic PAS problem, only elective patients whose surgeries can be planned in advance are considered, and all patient admission and discharge requirements are deterministic (Demeester et al., 2010). (Mixed) integer programming (MIP/IP) models are generally used to formulate the classic PAS problem. Various methods have been proposed to solve this problem, such as the hybrid tabu search (H-TS) (Demeester et al., 2010), warm-start (WS) (Bastos et al., 2019), and constraint aggregation (CA) (Liu et al., 2024). Recently, Brandt et al. (2025b) developed a greedy heuristic (GH) to solve the integrated patient-to-room and nurse-to-patient assignment problem. A review of the classic PAS problem can be found in Liu et al. (2024).

For the uncertain operational-level PAS problem, the PASU problem extends the classic PAS by considering several real-world features, such as uncertain arrivals of urgent and emergency patients, delayed admissions, and LOS uncertainty (Ceschia and Schaerf, 2012). However, previous studies mainly focus on the uncertainty of patient arrivals, handling overstays by predefining a set of potential overstay patients with fixed estimated lengths of stay. The admission plan is periodically adjusted in response to new events, such as the admission of new patients or the overstay of existing ones. For solving PASU, a sequence of static problems is gradually solved as new information becomes available. Simulated annealing (SA) (Ceschia and Schaerf, 2012), adaptive large neighborhood search (ALNS) (Lusby et al., 2016), a general solver (Vancroonenburg et al., 2016) are generally

applied. Moreover, Ceschia and Schaerf (2016) extended the PASU problem to incorporate operating room resource constraints and proposed a SA algorithm to solve this problem. Later, Zhu et al. (2019) proposed a MIP model, and developed a Dantzig-Wolfe decomposition (DWD) and column generation (CG) algorithm to solve this problem. Guido (2024) addressed the PASU problem as an offline problem where all patients are known and all the needed information is available. A matheuristic algorithm, named FiNeMath-PASU, that combines the large neighborhood search with the Fix-and-optimize heuristic is proposed.

In addition to the PASU studies, several works explored healthcare scheduling problems that incorporate patient admission decisions. However, these studies typically focus on room capacity and admission time windows constraints, while ignoring many of the soft and hard constraints that are central to PASU. For instance, Dai and Shi (2019) examined the overflow decision problem, proposing a Markov decision process (MDP) model and developing an approximate dynamic programming (ADP) algorithm to solve it. Their work primarily focuses on bed capacity and overflow management, without incorporating constraints related to room assignments and patient preferences. Jiang et al. (2023) addressed the admission decision of patients who should immediately be admitted into a same-gender room or rejected. They formulated the problem as an MDP model to describe the stochastic arrivals of patients, and solved it using a value iteration (VI) algorithm. For a comprehensive review of MDP-based models in patient admission scheduling, we refer the reader to Jiang et al. (2023). Gartner and Kolisch (2014) studied the hospital-wide patient scheduling (HWPS) problem, taking into account diagnostic resources, operating rooms, and beds. They proposed MIP models, where the LOS of patients are considered as decision variables in their models instead of parameters, and embedded the models into a static and a RHP approach. Vancroonenburg et al. (2019) developed a chance-constrained (CC) model to determine admission and surgery scheduling decisions of elective surgical patients. The CC model was implemented by using SAA and solved by late acceptance hill climbing (LAHC) algorithm. Brandt et al. (2025a) presented combinatorial insights about the feasibility of patient-to-room assignment (PRA), proposed different IP formulations for PRA and embedded them into RHP for solving the dynamic PRA.

Although PAS and operating room (OR) scheduling (Neyshabouri and Berg, 2017; Shehadeh et al., 2026) are closely related, they focus on different operational decision problems.

Existing OR studies prioritize surgical performance, such as minimizing overtime, and typically treat downstream bed occupancy as a secondary constraint. In contrast, our work directly targets patient admission and patient-room assignment decisions, with bed occupancy as the central operational concern. Moreover, unlike OR scheduling studies that typically model LOS as fully stochastic, we adopt a fixed LOS plus stochastic overstay.

In summary, our study builds on the classical PAS problem and extends it by modeling stochastic overstays using a state-variable modeling approach and integrating the model into a RHP framework to handle dynamic patient arrivals. Table 1 provides a summary of the research on the SPAS problem along with the uncertainty considered, the scheduling type, problem constraints, mathematical model and algorithm. We indicate both the offline and online decision-making, specify which constraints are considered as hard or soft in each study, and identify model type and the LOS modeling approach. The symbols "✓" and "-" are used to indicate whether a study considers a specific aspect or not. "Δ" indicates that the online decision-making approach is an extension of the proposed offline method.

Table 1: A summary of studies related to the SPAS problem

Reviewed literature	Uncertainty	Scheduling type		Problem constraints		Mathematical Model		Algorithm
		Offline	Online	Hard	Soft	Model type	LOS modeling	
Demeester et al. (2010)	-	✓		1, 3-5, 7, 10	3-8, 10	IP	ELOS	H-TS
Bastos et al. (2019)	-	✓		1, 5, 7, 10	3-8, 10	MIP	ELOS	WS
Liu et al. (2024)	-	✓		1, 5, 7, 10	3-8, 10	IP	ELOS	WS, CA
Brandt et al. (2025b)	-	✓		1, 10	3-8, 10	MIP	ELOS	GH
Ceschia and Schaerf (2012)	LOS, PA		✓	1-3, 5-7	2, 4-9	IP	FLOS + DOVS	SA
Gartner and Kolisch (2014)	LOS, PA*	✓	Δ	1, 2, 5	10	MIP	OLOS	Solver, RHP
Ceschia and Schaerf (2016)	LOS, PA		✓	1-3, 5-7	2, 4-10	-	FLOS + DOVS	SA
Lusby et al. (2016)	LOS, PA		✓	1-3, 5-7	2, 4-9	MIP	FLOS + DOVS	ALNS
Vancroonenburg et al. (2016)	LOS, PA		✓	1, 5, 7, 10	3-8, 10	IP	FLOS + DOVS	Solver
Dai and Shi (2019)	LOS, PA		✓	5,10	-	MDP	SLOS	ADP
Vancroonenburg et al. (2019)	LOS, SD		✓	1, 2, 5, 10	10	CC	SLOS	SAA, LAHC
Zhu et al. (2019)	LOS, PA		✓	1-3, 5-7	2, 4-10	MIP	ELOS	DWD, CG
Jiang et al. (2023)	LOS, PA		✓	5	4	MDP	-	VI
Guido (2024)	LOS	✓		1-3, 5, 7, 8	2, 4-9	MIP	FLOS + DOVS	FiNeMath-PASU
Brandt et al. (2025a)	PA*	✓	Δ	1, 4, 5	8,10	IP	ELOS	Solver, RHP
This paper	LOS, PA*	✓	Δ	1-3, 5-8	2, 4-9	SB, SV	FLOS + SOVS	SAA-SV, RHP

Uncertainty: LOS - length of stay; SD - surgery duration; PA - (emergency) patient arrival; RC - resource capacity
Problem constraints: 1 - complete assignment; 2 - admission time window; 3 - age policies; 4 - gender policies; 5 - room capacity; 6 - specialisms; 7 - equipment; 8 - transfers; 9 - overcrowd risk; 10 - others
Model type: SB - scenario-based model; MIP - mixed integer programming model; DRO - distributionally robust optimization model MDP - Markov decision process model; SV - state-variable model; QM - queueing model
LOS modeling: SLOS - stochastic LOS; ELOS - expected LOS; SOVS - stochastic overstay; DOVS - deterministic overstay; FLOS - fixed LOS; OLOS - optimizable LOS
* These studies first developed a static method without considering PA, and subsequently embedded it into the RHP to cope with PA.

3 Problem description and stochastic programming model

3.1 Problem description

The definition and terminology of the SPAS problem, inspired by Ceschia and Schaerf (2012) and Guido (2024), are provided as follows.

Day: The smallest unit of time is a *day*, which is used to express the LOS of each patient in the hospital. The set of consecutive days considered in the problem is called the *planning horizon*.

Patient: Each patient must be assigned to a *bed* in a *room*. Each patient has an admitting time window which is a range of days allowed to be admitted. Each patient has a fixed LOS that must be met, and each patient may have a risk of extending his/her LOS, termed *overstay*, which introduces the possibility that the patient may need to spend more nights in the hospital. Note that Ceschia and Schaerf (2012) assumes that patients may spend one extra night in the hospital, while we consider the possibility of patients spending multiple extra nights, following Yang et al. (2010), which is more realistic. Patients can change rooms during their hospitalization, termed $transfers$.

Department/room: Each room belongs to a *department*, and each department is correlated with the specialisms it offers. There are four types of room capacity (typically one, two, four or six), and each patient has a preference for a certain type of room. Overstay patients may lead to bed shortages, and hospitals can use extra beds in rooms to accommodate new patients.

Specialism: Patients pursue medical treatments during their hospitalization, termed *specialism*. Departments might be fully, partially, or not qualified for the treatment of a disease. Each department and room have their own priority degree for those specialisms.

Room feature: Each room has a different available equipment, such as oxygen and telemetry. Patients may require or prefer to be allocated to a room with specific equipment depending on their treatment.

Gender policy: Each room has a *gender policy*. Rooms that require patients to be of the same gender enforce policy M (only Male) or policy F (only Female). In contrast, rooms where both genders are allowed enforce policy N (mixed gender) or policy D (on any given night, only patients of the same gender are allowed).

Age policy: Each department has an *age policy*, which imposes a maximum or minimum age limit for acceptance.

Based on the above problem definition, Table 2 presents a complete classification of the hard and soft constraints. Feasibility requires that each patient is assigned a bed such that no hard constraints (types HC1 - HC7) are violated. Conversely, the quality of a feasible solution is dictated by the seven types of soft constraints, where each violated soft constraint

penalizes the objective function by a positive integer. Within this framework, requirements such as necessary qualifications and essential features are enforced as hard constraints, while partial qualifications and preferred features are treated as soft constraints. For instance, age policy constitutes a hard constraint because certain wards (e.g., pediatrics or geriatrics) strictly restrict admission to specific age ranges, rendering any non-compliant assignment infeasible. In contrast, gender policy is modeled as a soft constraint, representing a desirable but not mandatory room feature that may be relaxed under capacity constraints.

Table 2 Hard and soft constraints

Constraint	Definition
Hard constraints	
HC1	*Patient admission* - Each patient can only be admitted within the day range of his/her admission time window.
HC2	*Consecutive nights* - The LOS of patient is continuous. Each patient must stay in the hospital for the LOS.
HC3	*Department specialism* - Patients must be treated at the departments where the specialisms they need are offered.
HC4	*Mandatory equipment* - A patient is assigned to a room with the required room properties for his/her treatment.
HC5	*Age policy* - Age policy must be satisfied for all rooms.
HC6	*Room capacity* - Room occupancy on each day must not exceed capacity unless additional beds are allocated to overstay patients, subject to extra bed usage constraints.
HC7	*No transfers* - Patients cannot be transferred during the overstay period because their extended stay typically indicates a more complex condition or slower recovery. They must still stay in the room assigned for their last planned night.
Soft constraints	
SC1	*Delayed admission* - The earlier the patient is admitted, the better.
SC2	*Gender policy* - Gender policy is satisfied for each room.
SC3	*Levels of expertise* - A patient is assigned to a department with the highest priority for his/her specialism.
SC4	*Preferred room capacity* - The room type preference for each patient is met.
SC5	*Preferred equipment* - A patient is assigned to a room with preferred room properties.
SC6	*Transfers* - A patient can be transferred from one room to another room over his/her fixed LOS to improve overall patient satisfaction by better matching room resources to patient needs and preferences. However, it is preferable to keep transfers to a minimum to avoid discomfort, maintain continuity of care.
SC7	*Extra bed usage* - Extra beds are allowed only in rooms with existing overstay patients, and their number cannot exceed a specified threshold.

The optimization objective of the SPAS problem is to find a feasible assignment satisfying constraints HC1 - HC7, while minimizing the expected total cost associated with violations of soft constraints SC1-SC7 under uncertainty in patient overstays.

3.2 Scenario-based stochastic programming model

To formulate the SPAS problem, we consider it as a two-stage SP problem. The first stage decides the admission date of each patient from the allowed dates and assigns suitable rooms for their fixed LOS. The second stage computes the expected recourse cost under stochastic overstays. For each scenario, feasibility and penalties are determined by patient occupancy, including costs associated with overstay assignments, extra-bed usage, and gender-policy violations. Although no first-stage decisions are revised, the second stage checks the feasibility of the first-stage decisions and involves a structured evaluation of scenario-dependent constraints and costs. Thus, we propose a scenario-based SP model $SPAS_{SB}$, where each scenario represents a possible realization of the overstay lengths for all patients. The $SPAS_{SB}$ model, with notations in Table 3, is defined as follows:

Table 3 Notation used for the $SPAS_{SB}$ model.

Symbol	Description
Sets	
$\mathcal{P}$	Set of patients ($p = 1, \cdots, \lvert\mathcal{P}\rvert$)
$\mathcal{D}$	Set of days ($i, d = 1, \cdots, \lvert\mathcal{D}\rvert$)
$\mathcal{R}$	Set of rooms ($r = 1, \cdots, \lvert\mathcal{R}\rvert$)
Ω	Set of scenarios ($\omega = 1, ..., \lvert\Omega\rvert$)
$\mathcal{P}^M \subset \mathcal{P}$	Set of male patients
$\mathcal{P}^F \subset \mathcal{P}$	Set of female patients
$\mathcal{R}^D \subset \mathcal{R}$	Set of rooms following gender policy D, referred to as dependent rooms
$\mathcal{R}_p \subset \mathcal{R}$	Set of rooms can be assigned to patient p without violating the hard constraints HC3 - HC5
$\mathcal{D}_p^A \subset \mathcal{D}$	Set of admission dates of patient p ($d = D_p^{EA}, ..., D_p^{LA}$, where D_p^{EA} is the earliest admission date and D_p^{LA} is the latest admission date)
$\mathcal{D}_p \subset \mathcal{D}$	Set of dates of patient p who may stay in the hospital ($d = D_p^{EA} + 1, ..., D_p^{LA} + L_p - 1$)
$\mathcal{D}_{pi}^L \subset \mathcal{D}$	Set of hospitalization dates of patient p who is admitted on day i ($d = i, ..., i + L_p - 1$)
$\bar{\mathcal{D}}_{pd}^A \subset \mathcal{D}$	Set of valid admission days for patient p such that day d falls within their hospitalization dates, i.e., $\bar{\mathcal{D}}_{pd}^A = \{i \in \mathcal{D}_p^A \mid d \in \mathcal{D}_{pi}^L\}$.
$\widetilde{\mathcal{D}}_{pi\omega} \subset \mathcal{D}$	Set of overstay dates of patient p who is admitted on day i ($d = i + L_p, ..., i + L_p + \widetilde{L}_{p\omega} - 1$)
Parameters	
G_p	Gender of patient p (M or F)
B_r	Number of beds in room r
L_p	Fixed length of stay of patient p
$\widetilde{L}_{p\omega}$	Length of overstay of patient p under scenario ω
C_{pr}	Penalty of assigning patient p to room r. The room penalties of SC2 (gender policy M, F, N), SC3 - SC5 are incorporated into the value
W^{RG}	Penalty weight for violating the gender policy constraint (gender policy D)
W^{Tr}	Penalty weight for patient transfers
W^{OP}	Penalty weight for using extra beds
W^{De}	Penalty weight for delayed patient admission
O_r	Maximum number of extra beds allowed in room r
$\Pr(\omega)$	Probability of scenario ω
λ_{rd}^F, λ_{rd}^M	Positive numbers
First-stage decision variables	
α_{pi}	1 if patient p is admitted on day i, 0 otherwise
x_{prid}	1 if patient p is admitted on day i and assigned to room r on day d, 0 otherwise
t_{pd}	1 if patient p is transferred on day d, 0 otherwise
Second-stage decision variables (recourse decisions)	
$u_{rd\omega}$	1 if there is at least one female patient in room r on day d under scenario ω, 0 otherwise
$b_{rd\omega}$	1 if there are both male and female patients in room r on day d under scenario ω, 0 otherwise
$z_{rd\omega}$	Number of extra beds used in room r on day d under scenario ω

$$\boldsymbol{SPAS_{SB}}: \text{Min } Cost_{SB} = \sum_{p\in\mathcal{P}} \sum_{i\in\mathcal{D}_p^A} W^{De}(i - D_p^{EA})\alpha_{pi} + \sum_{p\in\mathcal{P}} \sum_{r\in\mathcal{R}_p} \sum_{i\in\mathcal{D}_p^A} \sum_{d\in\mathcal{D}_{pi}^L} C_{pr}x_{prid} + \sum_{p\in\mathcal{P}|L_p\geq 2} \sum_{d\in\mathcal{D}_p} W^{Tr}t_{pd} + \sum_{\omega\in\Omega} \Pr(\omega)Q_1(\boldsymbol{x},\omega) \quad (1)$$

$$\text{s.t.} \quad \sum_{i\in\mathcal{D}_p^A} \alpha_{pi} = 1 \quad \forall p \in \mathcal{P} \quad (2)$$

$$\sum_{r\in\mathcal{R}_p} x_{prid} = \alpha_{pi} \quad \forall p \in \mathcal{P}, i \in \mathcal{D}_p^A, d \in \mathcal{D}_{pi}^L \quad (3)$$

$$\sum_{p\in\mathcal{P}|r\in\mathcal{R}_p}\sum_{i\in\mathcal{D}_p^A|d\in\mathcal{D}_{pi}^L} x_{prid} \le B_r \quad \forall r\in\mathcal{R}, d\in\mathcal{D} \tag{4}$$

$$t_{pd} \ge \sum_{i\in\bar{\mathcal{D}}_{pd}^A} x_{prid} - \sum_{i\in\bar{\mathcal{D}}_{p,d-1}^A} x_{pri,d-1} - \alpha_{pd}$$
$$\forall p\in\mathcal{P}|L_p\ge 2, r\in\mathcal{R}_p, d = D_p^{EA}+1,...,D_p^{LA} \tag{5}$$

$$t_{pd} \ge \sum_{i\in\bar{\mathcal{D}}_{pd}^A} x_{prid} - \sum_{i\in\bar{\mathcal{D}}_{p,d-1}^A} x_{pri,d-1}$$
$$p\in\mathcal{P}|L_p\ge 2, r\in\mathcal{R}_p, d = D_p^{LA}+1,\ldots,D_p^{LA}+L_p-1 \tag{6}$$

$$\alpha_{pi}\in\{0,1\} \quad \forall p\in\mathcal{P}, i\in\mathcal{D}_p^A \tag{7}$$

$$x_{prid}\in\{0,1\} \quad \forall p\in\mathcal{P}, r\in\mathcal{R}_p, i\in\mathcal{D}_p^A, d\in\mathcal{D}_{pi}^L \tag{8}$$

$$t_{pd}\in\{0,1\} \quad \forall p\in\mathcal{P}|L_p\ge 2, d\in\mathcal{D}_p \tag{9}$$

where

$$Q_1(\boldsymbol{x},\omega) = \text{Min}\sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}_p}\sum_{i\in\mathcal{D}_p^A}|\widetilde{\mathcal{D}}_{pi\omega}|C_{pr}x_{pri,i+L_p-1} + \sum_{r\in\mathcal{R}^D}\sum_{d\in\mathcal{D}}W^{RG}b_{rd\omega} + \sum_{r\in\mathcal{R}}\sum_{d\in\mathcal{D}}W^{OP}z_{rd\omega} \tag{10}$$

$$\text{s.t.} \quad z_{rd\omega} \ge \sum_{p\in\mathcal{P}|r\in\mathcal{R}_p}\sum_{i\in\mathcal{D}_p^A}\left[\mathbb{1}(d\in\mathcal{D}_{pi}^L)x_{prid} + \mathbb{1}(d\in\widetilde{\mathcal{D}}_{pi\omega})x_{pri,i+L_p-1}\right] - B_r$$
$$\forall r\in\mathcal{R}, d\in\mathcal{D}, \omega\in\Omega \tag{11}$$

$$\lambda_{rd}^F(u_{rd\omega}+b_{rd\omega}) \ge \sum_{p\in\mathcal{P}^F|r\in\mathcal{R}_p}\sum_{i\in\mathcal{D}_p^A}\left[\mathbb{1}(d\in\mathcal{D}_{pi}^L)x_{prid} + \mathbb{1}(d\in\widetilde{\mathcal{D}}_{pi\omega})x_{pri,i+L_p-1}\right]$$
$$\forall r\in\mathcal{R}^D, d\in\mathcal{D}, \omega\in\Omega \tag{12}$$

$$\lambda_{rd}^M(1-u_{rd\omega}+b_{rd\omega}) \ge \sum_{p\in\mathcal{P}^M|r\in\mathcal{R}_p}\sum_{i\in\mathcal{D}_p^A}\left[\mathbb{1}(d\in\mathcal{D}_{pi}^L)x_{prid} + \mathbb{1}(d\in\widetilde{\mathcal{D}}_{p\omega})x_{pri,i+L_p-1}\right]$$
$$\forall r\in\mathcal{R}^D, d\in\mathcal{D}, \omega\in\Omega \tag{13}$$

$$b_{rd\omega}\in\{0,1\} \quad \forall d\in\mathcal{D}, r\in\mathcal{R}^D, \omega\in\Omega \tag{14}$$

$$u_{rd\omega}\in\{0,1\} \quad \forall d\in\mathcal{D}, r\in\mathcal{R}^D, \omega\in\Omega \tag{15}$$

$$z_{rd\omega}\in[0,O_r] \quad \forall d\in\mathcal{D}, r\in\mathcal{R}, \omega\in\Omega \tag{16}$$

In the first stage, the objective function (1) minimizes violations of the soft constraints. Each soft constraint has an associated penalty that is incurred if it is violated. The first part calculates the delayed admissions with respect to the earliest admission date, related to soft constraint SC1. The second part computes the cost of assigning patients to rooms, which is determined by the combined penalty of soft constraints SC2 (gender policy M, F, N), SC3, SC4 and SC5. The third part captures the cost associated with patient transfer related to soft constraint SC6. The last part of the function $Q_1(\boldsymbol{x},\omega)$ is the second-stage recourse function. Constraint (2) ensures that each patient is admitted only once and within the range of admission dates. Constraint (3) ensures each patient is assigned to a room for every day of their hospitalization based on the assigned admission date. Constraint (4)

ensures the number of patients assigned to each room does not exceed the room capacity. Constraints (5)-(6) ensure that the variable t_{pd} becomes 1 if a patient changes room on two consecutive days. Constraints (7)-(9) define the domains of the first-stage variables.

In the second stage, the objective function (10) minimizes the expected costs, including penalties for assigning patients to rooms within the overstay period, penalties for extra bed usage and for gender policy violations. Constraint (11) calculates the number of extra beds used in each room on each day under every scenario. In this constraint, $\mathbb{1}(\cdot)$ is an indicator function that is equal to 1 if condition $(\cdot)$ is satisfied, and 0 otherwise. For example, the condition of $\mathbb{1}(d \in \mathcal{D}^L_{pi})$ is that the day d falls within the hospitalization period of patient p admitted on day i (i.e., $d \in [i, i + L_p - 1]$). Constraints (12) and (13) refer to gender policy constraints, where $\lambda^F_{rd} = \min\{O_r + B_r, |\mathcal{P}^F_{rd}|\}$ and $\lambda^M_{rd} = \min\{O_r + B_r, |\mathcal{P}^M_{rd}|\}$, $|\mathcal{P}^F_{rd}|$ and $|\mathcal{P}^M_{rd}|$ are the number of female/male patients who can be assigned to room r on day d without violating hard constraints HC3, HC4 and HC5. These constraints are formulated based on the work of Liu et al. (2024). Specifically, constraint (12) enforces female patient restrictions, and constraint (13) enforces male patient restrictions. Both constraints seek to avoid the assignment of two distinct genders to the same room, penalizing allocations where different genders share a room. Constraints (14)-(16) define the domains of the second-stage variables (recourse decisions).

4 State-variable modeling and solution method

The $SPAS_{SB}$ model estimates the second-stage expected costs by enumerating a large number of scenarios, causing the model to be difficult to solve as it contains a large number of second-stage constraints and variables for these scenarios. Suppose that there are N patients to be admitted in the given planning horizon, and each patient may need to spend a maximum number of M extra nights in the hospital. Then, the number of scenarios is $(M + 1)^N$, where 1 represents the patient leaving the hospital on the scheduled discharge day. The number of scenarios will increase exponentially if the number of patients is large. To overcome this challenge, this section presents the state-variable modeling approach to reformulate the $SPAS_{SB}$ model, which allows for the computation of the second-stage expected costs without relying on explicit scenario enumeration.

4.1 Main idea of the state-variable modeling on SPAS problem

The main idea of reformulating the $SPAS_{SB}$ model via state-variable modeling is to represent the second-stage SPAS problem by replacing scenario-dependent decisions with a sequence of state transitions that describe the system evolution over time. To achieve this, we identify the decomposition structure of the second-stage SPAS problem, which reveals the local nature of the recourse decisions, and reformulate the recourse problem in a state-variable modeling framework. This subsection provides a conceptual overview of the reformulation logic, while the detailed construction is presented in the subsequent sections.

A critical observation is that the second-stage SPAS problem admits a natural decomposition by room-day pairs. In $SPAS_{SB}$, the second-stage feasibility and costs are triggered locally at the level of individual rooms on individual days: extra bed usage, and gender-policy violations all depend on how many and which patients occupy a specific room on a specific day. Conditional on the first-stage decisions x_{prid}, the recourse cost $Q_1(\boldsymbol{x}, \omega)$ incurred on one room-day pair is independent of occupancy outcomes in other rooms or on other days. Accordingly, the second-stage expected cost is equal to the sum of the contributions across all room-day pairs.

State-variable modeling thus can be introduced to characterize the patient overstay uncertainty associated with each room-day pair of the second-stage SPAS problem. Following the state-variable modeling framework (Doulabi et al., 2022), we decompose the second stage into a sequence of artificial decision stages by conceptually ordering the patients' LOS realizations (comprising both fixed LOS and potential overstays) to form a sequential state-transition process. For each room-day pair, a state is defined to capture the bed occupancy and gender status after processing each patient's stay. State transitions then characterize the evolution of these conditions as patients are sequentially considered (Section 4.2 provides the formal construction). By enumerating all reachable states at each period and applying a recursive cost function, we can evaluate the expected recourse cost through a series of incremental updates. Significantly, while the number of scenarios grows exponentially with the patient sequence (i.e., $(M+1)^N$), the overall scale of our formulation is governed by a compactly bounded space of generated periods, reachable room states per period, and valid admission windows. This fundamental shift from exponential to constant scaling makes the state-variable modeling approach significantly more com-

putationally efficient and tractable than traditional scenario-based modeling. A detailed comparison of the model scales and computational performance is provided at the end of Section 4. Thus, this approach effectively transforms the complex second-stage recourse into a tractable sequential decision-making problem.

Building on this idea, we introduce a stepwise evaluation process (SEP), as detailed in Section 4.2, to extract the key elements — period, state, action, transitions, and cost function — required by the state-variable modeling approach. Then, we develop the $SPAS_{SV}$ model in Section 4.3, where state variables summarize the intermediate room-day condition (e.g., bed occupancy and room gender), while state transitions capture its evolution as patients are processed sequentially. For a detailed presentation of the state-variable modeling framework and its theoretical foundations, we refer readers to Doulabi et al. (2022), where the paradigm is illustrated using a R&D project scheduling problem and the transformation from a scenario-based formulation to a state-variable model is explicitly demonstrated. Finally, Section 4.4 presents the SAA-SV method to solve the $SPAS_{SV}$ model efficiently.

4.2 Stepwise evaluation process for the 2-stage SPAS problem

4.2.1 Main scheme

To evaluate the expected costs of the second-stage SPAS problem, we decompose it into $|\mathcal{D}| \times |\mathcal{R}|$ independent subproblems, each corresponding to a specific room-day pair (r, d). The objective value of each subproblem is independently evaluated via a SEP. To initiate the SEP, we first identify the set of patients who may occupy room r on day d, denoted as $\hat{\mathcal{P}}_{rd}$. $\hat{\mathcal{P}}_{rd} = \{p \in \mathcal{P} \mid r \in \mathcal{R}_p, \exists i \in \mathcal{D}_p^A \text{ such that } d \in \mathcal{D}_{pi}^L \cup \hat{\mathcal{D}}_{pi}\}$, where $\hat{\mathcal{D}}_{pi} = \{d | d = i + L_p, ..., i + L_p + \max_{\omega \in \Omega} \widetilde{L}_{p\omega} - 1\}$ represents the potential overstay days for patient p admitted on day i. For each room-day pair (r, d), the SEP consists of a sequence of periods $\tau \in \mathcal{T}_{rd} (\mathcal{T}_{rd} = \{0, 1, ..., |\hat{\mathcal{P}}_{rd}|\})$. At each period τ, the assignment for the next patient $p_{\tau+1}$ (referred to as an action) is executed, the resulting room states (bed occupancy and room gender) are updated, along with the associated cost evaluation. The expected cost of the SEP is finally computed through a recursive cost function, which captures the expected costs incurred at each period. The key elements of the SEP are defined as follows.

- The room state at each period τ is $s_\tau^{rd} = (n_\tau^{rd}, g_\tau^{rd})$ if $(r \in \mathcal{R}^D)$, and $s_\tau^{rd} = n_\tau^{rd}$ other-

wise. Specifically, $n_\tau \in \mathcal{N}_{\tau rd}$ is the number of occupied beds, $\mathcal{N}_{\tau rd} = \{0, 1, ..., \min(\tau, B_r + O_r)\}$ is the set of all possible values for the number of occupied beds used by any patient 0 to p_τ. Here, patient 0 is a dummy patient, meaning that there is no patient assigned to room r on day d at the beginning of the SEP. $g_\tau \in \mathcal{G}_{\tau rd}$ is the gender occupancy state in room r on day d, where $\mathcal{G}_{\tau rd}$ is the set of all possible room gender state for a dependent room r ($r \in \mathcal{R}^D$) after assigning any subset of patients 0 to p_τ to that room on day d, $\mathcal{G}_{\tau rd} = \{None, \bigcup_{j=1}^{\tau} G_{p_j}\}$. $None$ means that no patient is assigned. The expression $\bigcup_{j=1}^{\tau} G_{p_j}$ can result in gender states including $OnlyM$, $OnlyF$, $Mixed$, corresponding to rooms occupied by male patients only, female patients only, or by both genders, respectively.

• The action a_τ^{rd} at period τ is reserving or not reserving a bed for the next patient $p_{\tau+1}$ in room r on day d, based on the first-stage decision variable x_{prid}. It is defined as an aggregated room-day reservation indicator over all possible admission days, $a_\tau^{rd} = \sum_{i \in \mathcal{D}^A_{p_{\tau+1}}} x_{p_{\tau+1}rid}$. Specifically, $a_\tau^{rd} = 1$ indicates that a bed is reserved for patient $p_{\tau+1}$, while $a_\tau^{rd} = 0$ indicates no reservation. When $a_\tau^{rd} = 1$, the corresponding variable $x_{p_{\tau+1}rid} = 1$ specifies the admission day i used in the subsequent state transition and cost evaluation.

• The state transition is the process of updating the room state from $s_\tau^{rd} = (n_\tau^{rd}, g_\tau^{rd})$ to $s_{\tau+1}^{rd} = (n_{\tau+1}, g_{\tau+1})$ based on the action a_τ^{rd} and the status of patient $p_{\tau+1}$ (whether he/she is discharged or not). Let $\Pr(p, l)$ be the probability distribution characterizing the LOS uncertainty of patient p, defined as the probability that patient p remains hospitalized on the l-th day after admission. $\Pr(p, l)$ can be obtained from the historical data and is assumed to be known. Thus, the state transition for $n_\tau^{rd} \to n_{\tau+1}^{rd}$ follows Eq. (17).

$$n_{\tau+1}^{rd} = \begin{cases} n_\tau^{rd}, & \text{if } \sum_{i \in \mathcal{D}^A_{p_{\tau+1}}} x_{p_{\tau+1}rid} = 0, \\ n_\tau^{rd} + 1, & \text{with probability } \Pr(p_{\tau+1}, d-i+1), \text{if } x_{p_{\tau+1}rid} = 1, \\ n_\tau^{rd}, & \text{with probability } 1 - \Pr(p_{\tau+1}, d-i+1), \text{ if } x_{p_{\tau+1}rid} = 1. \end{cases} \tag{17}$$

The transition for $g_\tau^{rd} \to g_{\tau+1}^{rd}$ follows Eq. (18), and is similar to $n_\tau^{rd} \to n_{\tau+1}^{rd}$.

$$g_{\tau+1}^{rd} = \begin{cases} g_\tau^{rd}, & \text{if } \sum_{i \in \mathcal{D}^A_{p_{\tau+1}}} x_{p_{\tau+1}rid} = 0, \\ g_\tau^{rd} \cup G_{p_{\tau+1}}, & \text{with probability } \Pr(p_{\tau+1}, d-i+1), \text{ if } x_{p_{\tau+1}rid} = 1, \\ g_\tau^{rd}, & \text{with probability } 1 - \Pr(p_{\tau+1}, d-i+1), \text{ if } x_{p_{\tau+1}rid} = 1. \end{cases} \tag{18}$$

• We define a recursive cost function to evaluate the SEP's expected cost. At period τ, for

a given $(s_\tau^{rd}, a_\tau^{rd})$ pair, the immediate expected cost is $C(s_\tau^{rd}, a_\tau^{rd})$. Thus, the expected cost from period τ to the terminal period $\hat{\mathcal{P}}_{rd}$ is given by:

$$V_\tau(s_\tau^{rd}) = C(s_\tau^{rd}, a_\tau^{rd}) + \mathbb{E}\left[V_{\tau+1}(s_{\tau+1}^{rd}) \mid s_\tau^{rd}, a_\tau^{rd}\right] \tag{19}$$

where $C(s_\tau^{rd}, a_\tau^{rd}) = C_{p'r}x_{p'ri,i+L_{p'}-1} + W_{rd\tau ni}^{EOP} + W_{rd\tau gi}^{ERG}$, $p' = p_{\tau+1}$. $C_{p'r}x_{p'ri,i+L_{p'}-1}$ represents the penalty cost for patient $p_{\tau+1}$ assigned to room r on his/her overstay day, where $i+L_{p'}-1$ is his/her last planned day after admission on day i. $W_{rd\tau ni}^{EOP}$ denotes the expected cost for potential extra bed usage caused by reserving a bed for patient $p_{\tau+1}$, admitted on day i to room r on day d, given that n beds are occupied (Eq. (20)). $W_{rd\tau gi}^{ERG}$ denotes the expected cost of violating the gender policy for reserving a bed for patient $p_{\tau+1}$, admitted on day i to room r on day d, given that room's gender occupancy state is g (Eq. (21)).

$$W_{rd\tau ni}^{EOP} = \begin{cases} W^{OP}\Pr(p_{\tau+1}, d-i+1), & \text{if } n_\tau^{rd} \geq B_r,\ x_{p_{\tau+1}rid} = 1 \\ 0, & \text{otherwise} \end{cases} \tag{20}$$

$$W_{rd\tau gi}^{ERG} = \begin{cases} W^{RG}\Pr(p_{\tau+1}, d-i+1), & \text{if } (g_\tau^{rd} = OnlyM \wedge G_{p_{\tau+1}} = F) \vee \\ & (g_\tau^{rd} = OnlyF \wedge G_{p_{\tau+1}} = M), x_{p_{\tau+1}rid} = 1 \\ 0, & \text{otherwise} \end{cases} \tag{21}$$

$\mathbb{E}[V_{\tau+1}(s_{\tau+1}^{rd}) \mid s_\tau^{rd}, a_\tau^{rd}]$ is the expected cost of future decisions, calculated based on the state transition probabilities resulting from the current state-action pair $(s_\tau^{rd}, a_\tau^{rd})$.

$$\mathbb{E}\left[V_{\tau+1}(s_{\tau+1}^{rd}) \mid s_\tau^{rd}, a_\tau^{rd}\right] = \begin{cases} \Pr(p_{\tau+1}, d-i+1)V_{\tau+1}\left(n_\tau+1, g_\tau \cup G_{p_{\tau+1}}\right) & \\ \quad + (1-\Pr(p_{\tau+1}, d-i+1))\, V_{\tau+1}(n_\tau, g_\tau), & \text{if } x_{p_{\tau+1}rid} = 1 \\ V_{\tau+1}(n_\tau, g_\tau), & \text{if } \sum_{i \in \mathcal{D}^A_{p_{\tau+1}}} x_{p_{\tau+1}rid} = 0 \end{cases} \tag{22}$$

Thus, $V_0(s_0^{rd})$ is the total expected cost of the SEP for room r on day d, with the initial state $s_0^{rd} = (0, None)$ if $r \in \mathcal{R}^D$, and $s_0^{rd} = 0$ otherwise. Although the recursive structure resembles a MDP, it is not a standard MDP since the cost parameters $C(s_\tau^{rd}, a_\tau^{rd})$ are dependent on the first-stage decisions x_{prid} and not fixed a priori. A detailed discussion of these differences is provided in Doulabi et al. (2022).

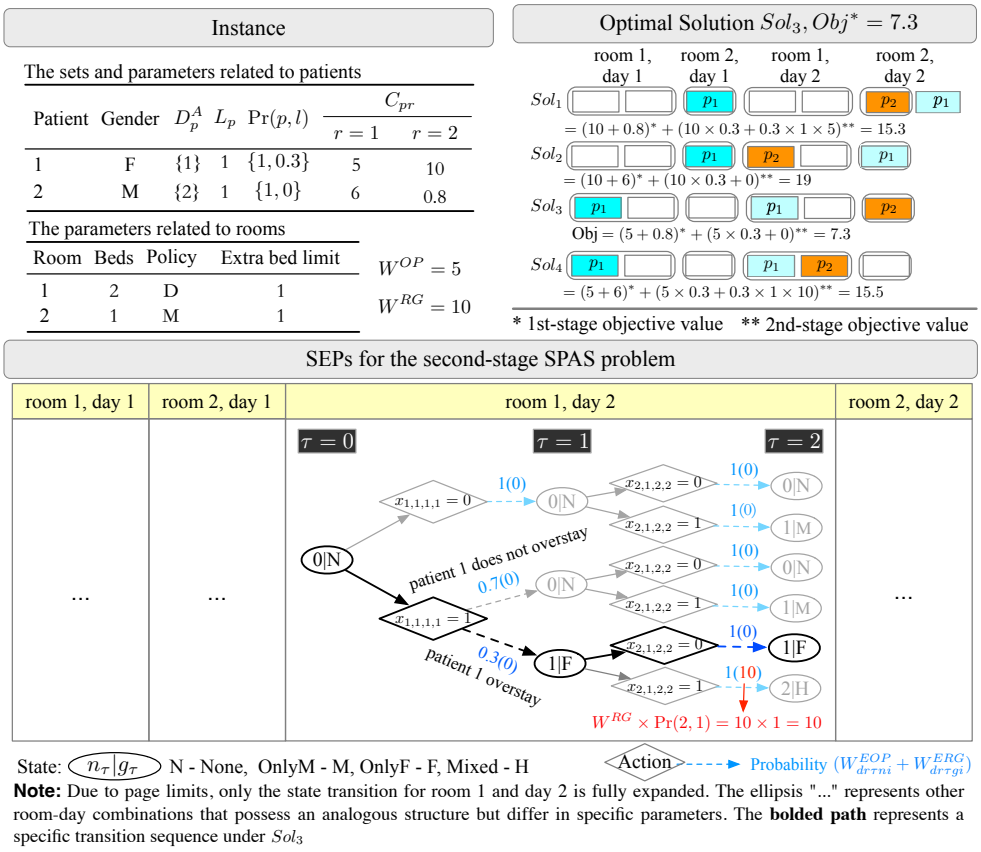
4.2.2 An illustrative example

To illustrate the SEP, consider an illustrative example of the SPAS problem with 2 patients, 2 rooms, and 2 days, as shown in Figure 1. Room 1 has 2 beds, and follows policy D.

Room 2 has 1 bed, and follows policy M. In this example, patient 1 must be admitted on day 1 and patient 2 must be admitted on day 2. Both patients have a planned LOS of 1 day ($\Pr(1,1) = Pr(2,1) = 1$), and patient 1 has a 30% chance of overstaying for an additional day ($\Pr(1,2) = 0.3$), while patient 2 will not overstay ($\Pr(2,2) = 0$). The table on the top left lists the input data related to the patients and rooms, where the meaning of the symbols corresponds to the definition mentioned in this section and Table 3. The lower part of the table lists four SEPs of the second-stage SPAS problem, where each SEP corresponds to a specific room-day pair. A state in these SEPs is represented as $(n_\tau|g_\tau)$, where n_τ is the number of occupied beds and g_τ denotes the gender occupancy state. As the SEPs for all room-day pairs share an analogous structure, Figure 1 only details the fully expanded transition for room 1 on day 2 for brevity. In this SEP, the potential patient set is $\hat{\mathcal{P}}_{1,2} = \{1, 2\}$, indicating that both patients may occupy room 1 on day 2. Consequently, the SEP is defined over three periods ($\tau = 0, 1, 2$) for each of which we illustrate the room state $s_\tau^{1,2}$, the action $a_\tau^{1,2}$ (governed by the first-stage variables $x_{1,1,1,1}$, $x_{1,2,1,1}$, $x_{2,1,2,2}$ and $x_{2,2,2,2}$), and the resulting state transitions.

Based on the constructed SEPs, we compute the expected cost associated with each feasible solution (Sol_1-Sol_4), as shown in the upper right section. For each solution, the corresponding formula for calculating the objective function value is provided immediately below. For example, for Sol_3, the 1st-stage objective value is the penalty cost related to $x_{1,1,1,1} = 1$ (patient 1 is admitted in day 1 and assigned to room 1 on day 1) and $x_{2,2,2,2} = 1$ (patient 2 is admitted in day 2 and assigned to room 2 on day 2), which is $C_{1,1} + C_{2,2} = 5 + 0.8$, and the 2nd-stage objective value can be calculated according to the SEP. Note that due to the decision of assigning patient 1 to room 1 and the HC7 (No transfers) requirement, patient 1 can only stay in room 1 on day 2 if he/she overstays. We denote a state transition by $\{s_\tau^{rd},\, a_\tau^{rd}\} \xrightarrow{\langle \text{event},\, C(s_\tau^{rd}, a_\tau^{rd})\rangle} \{s_{\tau+1}^{rd},\, a_{\tau+1}^{rd}\}$, where $\{s_\tau^{rd},\, a_\tau^{rd}\}$ and $\{s_{\tau+1}^{rd},\, a_{\tau+1}^{rd}\}$ are the state-action pairs at consecutive periods, *event* is the transition trigger, and $C(s_\tau^{rd}, a_\tau^{rd})$ is the immediate expected cost as defined in Eq.(19). For instance, on day 2 and room 1, a specific realization of the transition sequences for Sol_3 is illustrated by the bolded path: $\{s_0^{1,2}$: (0|N), $a_0^{1,2}$: reserve a bed for patient 1 (resulting from $x_{1,1,1,1} = 1)\}$ $\xrightarrow{\langle \text{patient 1 overstays with prob. 0.3, } C(s_0^{1,2}, a_0^{1,2})=1.5(\text{penalty } C_{1,1}=5 \text{ weighted by prob. 0.3})\rangle}$ $\{s_1^{1,2}$: (1(patient 1 is overstaying, he/she occupies a bed in room 1 on day 2, resulting in $n_1^{1,2} = 1$.)|F(patient

1 is a female, so the room gender state becomes $OnlyF$)), $a_1^{1,2}$: do not reserve a bed for patient 2 ($x_{2,1,2,2} = 0$)} $\xrightarrow{\langle \text{patient 2 stays with prob. 1, but is not assigned, } C(s_1^{1,2}, a_1^{1,2})=0 \rangle}$ {$s_2^{1,2}$: (1|F), $a_2^{1,2}$: None}. Here, all transitions along this path are consistent with the Eq.s.(17) and (18). This path captures the overstay scenario of patient 1, which occurs with a probability of 0.3 and contributes 1.5 to the expected cost. Conversely, the scenario where patient 1 leaves as planned ($1 - Pr(1, 2) = 0.7$) incurs zero cost. Thus, the objective value of Sol_3 is $5 + 0.8 + 5 \times 0.3 + 0 = 7.3$. Similarly, for other solutions, we can compute their objective values as $obj_{Sol_1} = 15.3$, $obj_{Sol_2} = 19$, and $obj_{Sol_4} = 15.5$. Therefore, the optimal solution is Sol_3 with an objective value of 7.3.



Note: Due to page limits, only the state transition for room 1 and day 2 is fully expanded. The ellipsis "..." represents other room-day combinations that possess an analogous structure but differ in specific parameters. The **bolded path** represents a specific transition sequence under Sol_3

Figure 1: Illustrative example of SEPs.

4.3 Mathematical Formulation of the $SPAS_{SV}$ Model

Using the elements defined in the SEP (period, state, action, transitions, and cost function), we next formulate the state-variable model $SPAS_{SV}$. This model utilizes state variables to track room states as each patient's hospitalization status unfolds sequentially. Additional

notation is provided in Table 4. Following the state-variable modeling approach, we introduce variables that represent the probability distribution of each (s_τ^{rd}, a_τ^{rd}). These variables are referred as state variables, each indexed by the corresponding state-action pair. Based

Table 4 Notation used for the $SPAS_{SV}$ model.

Symbol	Description
Sets	
$\hat{\mathcal{P}}^0_{rd}$	$\hat{\mathcal{P}}^0_{rd} = \hat{\mathcal{P}}_{rd} \cup \{0\}$, the index 0 is a dummy patient
$\mathcal{N}'_{\tau rd}$	Set of all possible values for the number of occupied beds by any patients 0 to p_τ who has a bed reserved in room r, assuming that there is a bed available for patient $p_{\tau+1}$ on day d ($n = 0, 1, ..., \min(\tau, B_r + O_r - 1)$). Note that the difference between $\mathcal{N}_{\tau rd}$ and $\mathcal{N}'_{\tau rd}$ lies that we ignore adding the future patient $p_{\tau+1}$ in $\mathcal{N}_{\tau rd}$. Conversely, we include $p_{\tau+1}$ in $\mathcal{N}'_{\tau rd}$ to ensure bed availability for this patient in room r on day d.
$\mathcal{G}^A_g$	Set of all possible predecessor states immediately prior to gender state g in the state transition process, i.e., $\mathcal{G}^A_{None} = \{None\}$, $\mathcal{G}^A_{OnlyM} = \{None, OnlyM\}$, $\mathcal{G}^A_{OnlyF} = \{None, OnlyF\}$, $\mathcal{G}^A_{Mixed} = \{OnlyF, OnlyM, Mixed\}$
State variables	
$y^1_{rd\tau ni}$	The probability of the number of occupied beds being equal to n on day d in room r at period τ, and we also reserve a bed in the same room for patient $p_{\tau+1}$ who is admitted on day i ($x_{p_{\tau+1},rid} = 1$)
$y^0_{rd\tau n}$	The probability of the number of occupied beds being equal to n on day d in room r at period τ, and we do not reserve a bed in the same room for patient $p_{\tau+1}$ ($\sum_{i\in\mathcal{D}^A_{p_{\tau+1}}} x_{p_{\tau+1},rid} = 0$)
$q^1_{rd\tau gi}$	The probability of room's gender occupancy state being g on day d in room r at period τ, and we also reserve a bed in the same room for patient $p_{\tau+1}$ who is admitted on day i ($x_{p_{\tau+1},rid} = 1$)
$q^0_{rd\tau g}$	The probability of room's gender occupancy state being g on day d in room r at period τ, and we do not reserve a bed in the same room for patient $p_{\tau+1}$ ($\sum_{i\in\mathcal{D}^A_{p_{\tau+1}}} x_{p_{\tau+1},rid} = 0$)

on the above elements and notation, we formulate the $SPAS_{SV}$ model as follows.

$$\begin{aligned}\boldsymbol{SPAS_{SV}}: \text{Min}\, Cost_{SV} = &\sum_{p\in\mathcal{P}}\sum_{i\in\mathcal{D}^A_p} W^{De}(i - D^{EA}_p)\alpha_{pi} + \sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}_p}\sum_{i\in\mathcal{D}^A_p}\sum_{d\in\mathcal{D}^L_{pi}} C_{pr}x_{prid} \\ &+ \sum_{p\in\mathcal{P}|L_p\geq 2}\sum_{d\in\mathcal{D}_p} W^{Tr}t_{pd} + Q_2(\boldsymbol{x}) \qquad (23)\\ \text{s.t.} \quad &\text{Constraints } (2) - (9)\end{aligned}$$

where

$$Q_2(\boldsymbol{x}) = \text{Min} \sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}_p}\sum_{i\in\mathcal{D}^A_p}\sum_{d\in\hat{\mathcal{D}}_{pi}} \Pr(p, d-i+1)C_{pr}x_{pri,i+L_p-1} \qquad (24)$$

$$\begin{aligned}&+\sum_{d\in\mathcal{D}}\sum_{r\in\mathcal{R}}\sum_{\tau\in\mathcal{T}_{rd}\setminus\{|\hat{\mathcal{P}}_{rd}|\}}\sum_{n\in\mathcal{N}'_{\tau rd}}\sum_{i\in\mathcal{D}^A_{p_{\tau+1}}} W^{EOP}_{rd\tau ni}y^1_{rd\tau ni} + \sum_{d\in\mathcal{D}}\sum_{r\in\mathcal{R}^D}\sum_{\tau\in\mathcal{T}_{rd}\setminus\{|\hat{\mathcal{P}}_{rd}|\}}\sum_{g\in\mathcal{G}_{\tau rd}}\sum_{i\in\mathcal{D}^A_{p_{\tau+1}}} W^{ERG}_{rd\tau gi}q^1_{rd\tau gi}\\ \text{s.t.} \quad &\sum_{i\in\mathcal{D}^A_{p_\tau}} \Big\{\mathbb{1}\left(n-1\in\mathcal{N}'_{\tau-1,rd}\right) y^1_{rd,\tau-1,n-1,i}\Pr(p_\tau, d-i+1)\\ &\qquad +\mathbb{1}\left(n\in\mathcal{N}'_{\tau-1,rd}\right) y^1_{rd,\tau-1,ni}\left[1-\Pr(p_\tau, d-i+1)\right]\Big\}\\ &+\mathbb{1}\left(n\in\mathcal{N}_{\tau-1,rd}\right) y^0_{rd,\tau-1,n} = \mathbb{1}\left(\tau\neq|\hat{\mathcal{P}}_{rd}|\wedge n\in\mathcal{N}'_{\tau rd}\right)\sum_{i\in\mathcal{D}^A_{p_{\tau+1}}} y^1_{rd\tau ni} + y^0_{rd\tau n}\end{aligned}$$

$$\forall d\in\mathcal{D}, r\in\mathcal{R}, \tau\in\mathcal{T}_{rd}\setminus\{0\}, n\in\mathcal{N}_{\tau rd} \qquad (25)$$

$$\sum_{i\in\mathcal{D}^A_{p_\tau}}\left\{\sum_{g'\in\mathcal{G}^A_g|g'\in\mathcal{G}_{\tau-1,rd}} q^1_{rd,\tau-1,g'i}\Pr(p_\tau,d-i+1)+\mathbb{1}\left(g\in\mathcal{G}_{\tau-1,rd}\right)q^1_{rd,\tau-1,gi}\left[1-\Pr(p_\tau,d-i+1)\right]\right\}$$
$$+\mathbb{1}\left(g\in\mathcal{G}_{\tau-1,rd}\right)q^0_{rd,\tau-1,g}=\mathbb{1}\left(\tau\neq|\hat{\mathcal{P}}_{rd}|\right)\sum_{i\in\mathcal{D}^A_{p_{\tau+1}}}q^1_{rd\tau gi}+q^0_{rd\tau g}$$
$$\forall d\in\mathcal{D},r\in\mathcal{R}^D,\tau\in\mathcal{T}_{rd}\setminus\{0\},g\in\mathcal{G}_{\tau rd} \tag{26}$$
$$\sum_{n\in\mathcal{N}'_{\tau rd}}y^1_{rd\tau ni}=\mathbb{1}\left(d\in\mathcal{D}^L_{p_{\tau+1}i}\right)x_{p_{\tau+1}rid}+\mathbb{1}\left(d\in\hat{\mathcal{D}}_{p_{\tau+1}i}\right)x_{p_{\tau+1}ri,i+L_{p_{\tau+1}}-1}$$
$$\forall d\in\mathcal{D},r\in\mathcal{R},\tau\in\mathcal{T}_{rd}\setminus\{|\hat{\mathcal{P}}_{rd}|\},i\in\mathcal{D}^A_{p_{\tau+1}} \tag{27}$$
$$\sum_{n\in\mathcal{N}_{\tau rd}}y^0_{rd\tau n}=1-\sum_{i\in\mathcal{D}^A_{p_{\tau+1}}}\left[\mathbb{1}\left(d\in\mathcal{D}^L_{p_{\tau+1}i}\right)x_{p_{\tau+1}rid}+\mathbb{1}\left(d\in\hat{\mathcal{D}}_{p_{\tau+1}i}\right)x_{p_{\tau+1}ri,i+L_{p_{\tau+1}}-1}\right]$$
$$\forall d\in\mathcal{D},r\in\mathcal{R},\tau\in\mathcal{T}_{rd}\setminus\{|\hat{\mathcal{P}}_{rd}|\} \tag{28}$$
$$\sum_{g\in\mathcal{G}_{\tau rd}}q^1_{rd\tau gi}=\mathbb{1}\left(d\in\mathcal{D}^L_{p_{\tau+1}i}\right)x_{p_{\tau+1}rid}+\mathbb{1}\left(d\in\hat{\mathcal{D}}_{p_{\tau+1}i}\right)x_{p_{\tau+1}ri,i+L_{p_{\tau+1}}-1}$$
$$\forall d\in\mathcal{D},r\in\mathcal{R}^D,\tau\in\mathcal{T}_{rd}\setminus\{|\hat{\mathcal{P}}_{rd}|\},i\in\mathcal{D}^A_{p_{\tau+1}} \tag{29}$$
$$\sum_{g\in\mathcal{G}_{\tau rd}}q^0_{rd\tau g}=1-\sum_{i\in\mathcal{D}^A_{p_{\tau+1}}}\left[\mathbb{1}\left(d\in\mathcal{D}^L_{p_{\tau+1}i}\right)x_{p_{\tau+1}rid}+\mathbb{1}\left(d\in\hat{\mathcal{D}}_{p_{\tau+1}i}\right)x_{p_{\tau+1}ri,i+L_{p_{\tau+1}}-1}\right]$$
$$\forall d\in\mathcal{D},r\in\mathcal{R}^D,\tau\in\mathcal{T}_{rd}\setminus\{|\hat{\mathcal{P}}_{rd}|\} \tag{30}$$
$$y^0_{rd\tau n}\in[0,1]\quad\forall d\in\mathcal{D},r\in\mathcal{R},\tau\in\mathcal{T}_{rd},n\in\mathcal{N}_{\tau rd} \tag{31}$$
$$y^1_{rd\tau ni}\in[0,1]\quad\forall d\in\mathcal{D},r\in\mathcal{R},\tau\in T_{rd}\setminus\{|\hat{\mathcal{P}}^0_{rd}|\},i\in\mathcal{D}^A_{p_{\tau+1}},n\in\mathcal{N}'_{\tau rd} \tag{32}$$
$$q^0_{rd\tau g}\in[0,1]\quad\forall d\in\mathcal{D},r\in\mathcal{R}^D,\tau\in\mathcal{T}_{rd},g\in\mathcal{G}_{\tau rd} \tag{33}$$
$$q^1_{rd\tau gi}\in[0,1]\quad\forall d\in\mathcal{D},r\in\mathcal{R}^D,\tau\in\mathcal{T}_{rd}\setminus\{|\hat{\mathcal{P}}^0_{rd}|\},i\in\mathcal{D}^A_{p_{\tau+1}},g\in\mathcal{G}_{\tau rd} \tag{34}$$

The first-stage $SPAS_{SV}$ model and the $SPAS_{SB}$ model is the same. Second-stage objective function (24) minimizes the expected cost of reserving beds for patients within the overstay period, and the penalties for extra bed usage and violation of gender policy. Constraints (25) and (26) are the state transition constraints, where constraint (25) links state n_τ to its previous states $n_{\tau-1}$ for room r on day d, and (26) links the state g_τ to its previous states $g_{\tau-1}$ for the same room and day. In both constraints, the left-hand side computes the transition probability based on the conditional probabilities from the previous period, while the right-hand side expresses the same probability using the state probabilities at the current period. Hence, each constraint enforces probability conservation and guarantees the correctness of the state transitions. For example, both sides of the constraint (25) independently represent the probability that the number of occupied beds at period τ is equal to n in room r on day d. The left-hand side properly calculates this probability value by condition probability relations based on the probabilities $y^1_{rd,\tau-1,ni}$, $y^1_{rd,\tau-1,n-1,i}$ and

$y^0_{rd,\tau-1,n}$. The right-hand side calculates this probability value by using the probabilities $y^1_{rd\tau ni}$ and $y^0_{rd\tau n}$. Each side of (25) independently and correctly computes the transition probability, ensuring flow balance across state transitions. Constraints (27) and (28) are the linking constraints between first-stage variables $x_{\tau+1,rid}$ and the state-variables $y^1_{rd\tau ni}$ and $y^0_{rd\tau n}$. Constraint (27) implies that if $x_{\tau+1,rid}$ is equal to 1 (or 0), then the sum of left-hand side state variables $y^1_{rd\tau ni}$ must be equal to 1 (or 0). Constraint (28) enforces the sum of left-hand side state variables $y^0_{rd\tau n}$ must be equal to 1 (or 0) if the sum of $x_{\tau+1,rid}$ is equal to 0 (or 1). Similarly, constraints (29) and (30) are the linking constraints between first-stage variables $x_{\tau+1,rid}$ and the state-variables $q^1_{rd\tau gi}$ and $q^0_{rd\tau g}$. Constraints (31) to (34) define the domains of the state variables.

It is worth noting that the patient sequence $\hat{\mathcal{P}}_{rd}$ can affect the SV model's performance (see Appendix A.5). In our study, we adopt a natural ordering based on ascending patient indices. This simple approach ensures consistency and avoids introducing additional heuristic biases, providing a clear baseline for evaluating the model's intrinsic efficiency. While the state-variable modeling approach was developed in Doulabi et al. (2022) based on an R&D project scheduling problem, our $SPAS_{SV}$ model adapts this approach to the SPAS problem. The resulting state variables $y^1_{rd\tau ni}$, $y^0_{rd\tau n}$, $q^1_{rd\tau gi}$, and $q^0_{rd\tau g}$ are different in structure and interpretation from the states considered in their application. Accordingly, we formally prove the validity of our $SPAS_{SV}$ model in Appendix D, as stated in Theorem 1.

Theorem 1 *The $SPAS_{SV}$ model is valid for the SPAS problem.*

4.4 Solution method for $SPAS_{SV}$ model

Although the $SPAS_{SV}$ model does not need to create variables for all possible scenarios, it is still difficult to solve the model directly due to a large number of variables and constraints. Considering that the SAA method can obtain a good approximate solution with relatively small sample scenarios, it is natural to apply the warm-starting strategy (De Santis et al., 2022) which uses SAA to obtain an approximate solution as the initial solution of the $SPAS_{SV}$ model. Therefore, our solution method SAA-SV first solves the $SPAS_{SAA}$ model to obtain an approximate solution and then solves the $SPAS_{SV}$ model using the obtained solution as the initial solution.

The SAA-SV method starts by initializing the necessary parameters and generating a

set of scenarios Ω^R, where Ω^R denotes the set of randomly generated scenarios used in the SAA procedure (See Appendix B.2). Then, it solves the $SPAS_{SAA}$ model within a time limit TL_{SAA} to obtain an approximate solution ($\boldsymbol{x}$, $\boldsymbol{t}$) (See Appendix A.1). To evaluate the objective value Obj of the solution ($\boldsymbol{x}$, $\boldsymbol{t}$) under the exponential number of scenarios, it fixes the values of the first-stage variables based on the solution ($\boldsymbol{x}$, $\boldsymbol{t}$) in the $SPAS_{SV}$ model and solves it. Once a better solution (with a lower objective value) is found, the best solution ($\boldsymbol{x_{SAA}}, \boldsymbol{t_{SAA}}$) of the SAA procedure along with its objective value Obj_{SAA} are updated. Since the optimal solution of the $SPAS_{SAA}$ model could be obtained before the time limit TL_{SAA} is reached, we utilize the remaining time TL_{SAA} - $elapsed_time$ to search for new solutions using a different set of scenarios Ω^R and launch a new round SAA procedure. Given that it is hard to improve the solution quality significantly in the subsequent iterations, we set a maximum number of iterations $MaxIter$ for the SAA procedure. The above SAA iterative procedure is repeated until either the time limit TL_{SAA} or the maximum number of iterations $MaxIter$ is reached. Note that for $Iter > 0$, we apply the warm-starting strategy which employs the best solution ($\boldsymbol{x_{SAA}}, \boldsymbol{t_{SAA}}$) as the initial solution of the new $SPAS_{SAA}$ model to accelerate the solution procedure. This is inspired from the iterative local search (Wang et al., 2023). After the SAA iterative procedure is completed, the $SPAS_{SV}$ model is solved using ($\boldsymbol{x_{SAA}}, \boldsymbol{t_{SAA}}$) as the initial solution within the remaining time. Our SAA-SV algorithm can be considered as a new solution framework for solving two-stage stochastic programming problems, featured by first producing a high-quality initial solution in a relatively short time and using it as a good upper bound (i.e. minimum problem) for pruning the search tree. The general scheme of the SAA-SV is presented in Algorithm 1 in Appendix B. Additionally, a comparison of the model sizes of the $SPAS_{SV}$ and $SPAS_{SAA}$ formulations is reported in Appendix B.3.

5 Computational Experiments

This section presents computational results to demonstrate the effectiveness of the $SPAS_{SV}$ model and the SAA-SV method, and to evaluate the stochastic programming approach for addressing the SPAS problem. We also present a real-life case study. Moreover, we examine the contribution of the warm-starting strategy in the SAA-SV method in Appendix F.4.

5.1 Instances design and experimental protocol

We carried out computational experiments on a total of 64 benchmark instances using the instance generator of Ceschia and Schaerf (2012). The generator receives the number of departments, rooms, features, patients, and days as input parameters. It creates a random instance based on their predefined distributions concerning various features such as the length of stay, the room capacity, the number of specialisms, etc. Note that Ceschia and Schaerf (2012) consider that the patients might need to spend one extra night in the hospital, which is a special case of our study. Thus, we adjusted their code (https://bitbucket.org/satt/PASU/) to allow patients to overstay for more than one night.

Table 5 shows the main characteristics of the benchmark instances. The instances are divided into 8 families, each of which is characterized by a different number of departments, rooms, equipment, patients, specialties and days. For each family, we generate 8 instances with different Demand-to-Supply Ratios (DSR) and different overstay lengths. The DSR is defined as the ratio of the total bed demand from all patients, excluding overstays, to the total number of beds available for all hospitalization days, which we set to be 40%, 50%, 60% and 70%. The overstay length is defined as the maximum number of days that a patient may overstay. The overstay length is set to be 1 and 2. Thus, we have 64 instances in total. All instances and code are publicly available (https://github.com/NWPU-ORMS/SPAS).

Table 5: Main characteristics of the benchmark instances.

Family of instances	Departments	Rooms	Equipment	Patients	Specialties	Days
Small short (S-S)	4	8	4	50	3	14
Small mid (S-M)	4	8	4	100	3	28
Small long (S-L)	4	8	4	200	3	56
Med short (M-S)	6	40	5	250	10	14
Med mid (M-M)	6	40	5	500	10	28
Med long (M-L)	6	40	5	1000	10	56
Large short (L-S)	8	160	6	1000	15	14
Large mid (L-M)	8	160	6	2000	15	28

Table 6 lists the cost setting used to penalize soft constraint violations, which is the same as Ceschia and Schaerf (2012); Lusby et al. (2016); Guido (2024) except for the weight of SC7. Moreover, for each room r, the maximum allowable number of extra beds O_r is set equal to B_r, which is provided by the instance generator and takes values in $\{1, 2, 4, 6\}$.

Our models were implemented and solved using Gurobi Optimizer 11.0.0 with its default

Table 6: Weights of the soft constraints.

Constraint	SC1	SC2	SC3	SC4	SC5	SC6	SC7
Weight	2	50	20	10	20	100	100

parameter settings. Experiments were run on a cluster with each node running Linux with Inter(R) Xeon(R) Gold 6226R 2.90GHz CPU and 256Gb RAM. The number of CPU cores used was set to be 1. For the S-S, S-M, S-L, M-S, and M-M instances, we set a time limit of 3600 seconds, whereas a time limit of 7200 seconds was used for the M-L, L-S, and L-M instances. We tested several sizes of the scenario set Ω^R with $|\Omega^R| = 5, 10, 20, 50$ for the SAA-SV method and found that $|\Omega^R| = 10$ yields a good trade-off between solution quality and efficiency (see Appendix F.2). The maximum number of iterations, $MaxIter$, is set to 3, and the maximum running time for solving the $SPAS_{SAA}$ model is limited to half of the total time limit.

5.2 Computational Results

As mentioned in Section 4, the number of scenarios increases exponentially with the number of patients, making the $SPAS_{SB}$ model difficult to solve. In our experiments, we found that all instances are intractable for the $SPAS_{SB}$ model due to insufficient memory errors. Thus, we do not report the computational results of the $SPAS_{SB}$ model. To simplify the expression, we refer to solving the $SPAS_{SV}$ model directly as the SV method in the following. Moreover, we directly solved a relaxed $SPAS_{SV}$ model within a maximum computation time of 24 hours to obtain lower bounds, which we refer to as the RSV method.

Table 7 summarizes the computational results among RSV, SV and SAA-SV methods on all the tested instances. For each method, we report the best lower bound obtained. For SV and SAA-SV methods, we also report the number of feasible solutions found, the best objective value found, the number of instances solved to optimality, and the average percentage gap (%). We compute the percentage gaps $Gap(\%) = (Obj - LB^*)/LB^* \times 100$ of the best objective value found by each method from the best lower bound LB^* among these three methods. Detailed results can be found in Appendix F.1. In terms of lower bounds, the SAA-SV, SV and RSV find the best LBs for 13, 17 and 11 instances, respectively. In terms of best objective values, the SAA-SV method dominates the SV method. Specifically, SAA-SV succeeds in finding feasible solutions for all 64 instances, while the SV method fails

for 31 large-scale instances. These two methods find the same best solutions for 8 small-scale instances, of which 3 and 5 instances are proven optimal solutions for SV and SAA-SV, respectively. Moreover, the averages of the percentage gaps for SV and SAA-SV methods are 40.0% and 2.2% for 1-day overstay instances, and 28.0% and 3.9% for 2-day overstay instances, respectively. Specifically, as the size of the instances increases, the obtained gaps range from 0.00% to 261.56% for SV and from 0.00% to 10.20% for SAA-SV. It is worth noting that SAA-SV finds high-quality feasible solutions with an average optimality gap of 3.75% for instances with 500 patients, 1-day overstay, and 3.27 $\times 10^{150}$ scenarios. Additionally, SAA-SV outperforms SV in terms of computational time. On average, for instances where both methods find feasible solutions with identical objective values, the SAA-SV is 2.67 times faster than SV. For instances where both methods reach the optimal solution, SAA-SV requires only 43% of the time taken by SV to prove optimality. These results suggest that incorporating SAA not only accelerates solution discovery but also improves the efficiency of optimality verification. To conclude, these results demonstrate that SAA-SV is quite effective in terms of both solution quality and computational time.

Table 7: Computational comparison among *RSV*, *SV* and *SAA-SV* methods

Instance set	RSV		SV					SAA-SV				
	Feasible Sol.	Best LB	Feasible Sol.	Best LB	Best Sol.	Solved optimally	Ave. Gap (%)	Feasible Sol.	Best LB	Best Sol.	Solved optimally	Ave. Gap (%)
1-day overstay (32 instances)	20	6	16	8	5	2	40.0	32	7	32	3	2.2
2-day overstay (32 instances)	20	5	17	9	3	1	28.0	32	6	32	2	3.9

5.3 Effect of stochasticity

We use the Value of Stochastic Solution (VSS) to measure the value gained by considering uncertain information when solving the problem with known distributions of random parameters. To compute the VSS, we first need to define the Expected Value Problem (EVP). Specifically, we create the EVP scenario by replacing the random overstay lengths with median-based overstay values, see Appendix B.2. Moreover, the SPAS problem aims to minimize a weighted sum of all the penalties of the unsatisfied soft constraints SC1-SC7. To sufficiently analyze the effect of stochasticity, we set the weight W^{OP} to be 100, 1000 and 2000. From a practical perspective, these values reflect a more realistic scenario where hospital managers prioritize minimizing the use of extra beds. From a modeling perspective, W^{OP} directly determines how uncertainty in patient overstays is traded off against

operational cost. By contrast, penalties relating to gender policy, patient transfers or delayed admissions reflect institutional or managerial preferences rather than being the main way in which stochasticity affects costs.

We present the results of the EVP and SAA-SV methods in Table 8. Detailed results can be found in Appendix F.3 (Tables 16 and 17). Each row in Table 8 corresponds to a specific instance set, and the columns provide the number of infeasible solutions for EVP and SAA-SV methods, as well as the average VSS values for each weight W^{OP}. The VSS value is computed as $VSS = (Obj_{EVP} - Obj_{SAA\text{-}SV})/SAA\text{-}SV \times 100$, where Obj_{EVP} and $Obj_{SAA\text{-}SV}$ are the objective values obtained from the EVP and SAA-SV methods, respectively. Moreover, to compute the true objective value Obj_{EVP}, we take the decisions obtained from the EVP model and evaluate them in the stochastic $SPAS_{SV}$ model. Since $SPAS_{SV}$ explicitly enforces capacity constraints across all scenarios, the EVP solution is deemed infeasible if it violates capacity even in a single realization during evaluation.

Table 8: Performance of EVP and $SAA\text{-}SV$ under varying W^{OP} values

Instance set	$W^{OP} = 100$			$W^{OP} = 1000$			$W^{OP} = 2000$		
	# Infeas. EVP	# Infeas. SAA-SV	Average VSS (%)	# Infeas. EVP	# Infeas. SAA-SV	Average VSS (%)	# Infeas. EVP	# Infeas. SAA-SV	Average VSS (%)
1-day overstay	0	0	3.86	0	0	24.73	0	0	45.03
2-day overstay	20(62.5%)	0	2.69	18(56.3%)	0	36.64	17(53.1%)	0	69.96

From Table 8, we observe that as the weight W^{OP} increases, the VSS values also increase. Specifically, the average VSS values are 3.86%, 24.73% and 45.03% when W^{OP} takes values of 100, 1000 and 2000, respectively. In addition, the VSS value tends to be higher for large-scale instances at a fixed DSR value, and also increases with larger DSR values for a given instance size. This suggests that the SAA-SV method is particularly well-suited for tackling large-scale problems with high DSR values, likely due to its effectiveness in managing the variability and uncertainty inherent in the SPAS. Moreover, we observe that the EVP method leads to feasible solutions for 44, 46, 47 instances when $W^{OP} = 100, 1000, 2000$ under the 2-day overstay, while the SAA-SV method can find feasible solutions for all instances. The infeasibility of solutions obtained by EVP in the $SPAS_{SV}$ model arises because the EVP model considers only one scenario, failing to capture all possible overstay days for each patient. As a result, capacity constraints are violated in some scenarios. For instances where EVP yields feasible solutions, the average

VSS values are 2.69%, 36.64% and 69.96% when the weight W^{OP} is 100, 1000 and 2000, respectively. This implies that SAA-SV outperforms EVP notably, especially under high penalties for extra bed utilization.

5.4 Integrating SAA-SV method within rolling horizon planning framework for solving real-life cases

Our previous study focused on dealing with the uncertainty in patient overstay lengths, assuming that all patients are known at the start of the planning horizon. In practice, new patients arrive dynamically and the hospital makes decisions on patient admissions and discharges daily. To enhance the practical applicability of our method, we embedded the SAA-SV method into a rolling horizon planning (RHP) framework (Gartner and Kolisch, 2014), called RHP & SAA-SV, to address the dynamic patient arrivals and discharges. Appendix C provides a detailed description of the RHP & SAA-SV method. In order to test the RHP & SAA-SV approach, we conducted a case study on the real-life dataset from a Chinese hospital, the People's Hospital of Xinjiang Uygur Autonomous Region. The characteristics of the LOS data for the selected departments are presented in Appendix E. The weights of the objective components are consistent with those used in Table 6. This real-life instance is also publicly available for future research work (https://github.com/NWPU-ORMS/SPAS). Moreover, we integrated the EVP and SAA methods in the RHP, resulting in two variants: RHP & EVP and RHP & SAA. The RHP is implemented in Python, and the computational experiments are conducted under the same environment as the previous tests.

To align with real-world scheduling strategies, we vary two key penalty parameters in the objective function: the penalty for patient transfers W^{Tr} which controls the amount of flexibility the hospital allows in moving patients across rooms, and the penalty for extra bed usage W^{OP} which controls the tolerance for using extra beds. Table 9 presents the actual objective values achieved by each method over a period of 30 days, decomposing the total solution costs into objective components. Each cost component is reported alongside relevant per-unit statistics (e.g., average delay per patient, total number of transfers, and total number of extra beds used), providing a more operationally interpretable view of model performance. From the table, we can observe that the RHP & SAA-SV consistently achieves the lowest objective values across all penalty configurations, indicating superior

performance. The results also reveal that the operational strategies vary significantly in how they handle overstay uncertainty. Specifically, for SAA-SV, under a low extra bed usage penalty ($W^{OP} = 100$), the models make greater use of extra beds (20 beds), helping reduce admission delays. When the extra bed usage penalty is increased ($W^{OP} = 1000$), extra bed usage drops to zero, and the hospital compensates with more patient transfers (59 transfers). Similarly, increasing the transfer penalty ($W^{Tr} = 1000$) leads to a sharp drop in transfer numbers (e.g., down to 21 in SAA-SV), but extra bed usage increases, illustrating a trade-off between the two strategies.

Table 9: Comparison of RHP with EVP, SAA, and SAA-SV on the real-life instance.

Approach	Total cost	Delay (d/p, n)	Trs. (t)	Ex. bed (b)	Gen. (v/r/d)	Spec. (m/p/d)	Room (m/p/d)	Equip. (m/p/d)
$W^{Tr} = 100$, $W^{OP} = 100$								
RHP&EVP	214,780	1,670 (1.34, 624)	2,800 (28)	4,800 (48)	186,050 (3.56)	12,880 (1.03)	5,680 (0.91)	900 (0.07)
RHP&SAA	214,014	1,804 (1.45, 624)	3,100 (31)	2,300 (23)	185,600 (3.56)	14,400 (1.15)	5,510 (0.88)	1,300 (0.1)
RHP&SAA-SV	213,654	1,724 (1.38, 624)	2,700 (27)	2,000 (20)	185,550 (3.55)	14,680 (1.18)	5,820 (0.93)	1,180 (0.09)
$W^{Tr} = 100$, $W^{OP} = 1,000$								
RHP&EVP	220,152	1,712 (1.37, 624)	7,100 (71)	0 (0)	186,850 (3.58)	16,240 (1.3)	6,170 (0.99)	2,080 (0.17)
RHP&SAA	218,956	1,706 (1.37, 624)	6,500 (65)	0 (0)	186,350 (3.57)	16,500 (1.32)	6,340 (1.02)	1,560 (0.12)
RHP&SAA-SV	215,348	1,648 (1.32, 624)	5,900 (59)	0 (0)	187,000 (3.58)	13,420 (1.08)	5,780 (0.93)	1,600 (0.13)
$W^{Tr} = 1,000$, $W^{OP} = 100$								
RHP&EVP	242,480	1,520 (1.22, 624)	30,000 (30)	4,800 (48)	186,700 (3.58)	12,740 (1.02)	5,380 (0.86)	1,340 (0.11)
RHP&SAA	236,792	1,812 (1.45, 624)	26,000 (26)	2,600 (26)	184,500 (3.53)	14,640 (1.17)	5,920 (0.95)	1,320 (0.11)
RHP&SAA-SV	230,990	1,770 (1.42, 624)	21,000 (21)	2,200 (22)	184,550 (3.54)	14,200 (1.14)	5,690 (0.91)	1,580 (0.13)

* Trs., Ex. bed., Gen, Spec, Room, Equip - the penalty related to transfers policy violations, extra beds, gender policy violations, not accounting for the prioritized specialism, failing to meet with his/her preferred capacity, failing to meet with his/her needed treatment properties.
** d/p = delay per patient, n = number of admitted patients, b = number of extra beds used, t = number of transfers, v/r/d = gender policy violations per room per day, m/p/d = mismatch per patient per day.

The results highlight several key managerial insights into hospital bed management. Patient overstays directly contribute to bed shortages, leading hospitals to rely on two primary coping strategies: patient transfers and extra bed usage. While both approaches provide flexibility in managing admission uncertainty, they often entail trade-offs in operational priorities. Reducing transfers increases reliance on extra beds and associated costs, while restricting extra bed usage leads to more frequent transfers, which can disrupt care continuity. These findings suggest that hospital administrators should carefully balance these strategies based on their specific resource constraints and operational objectives.

6 Conclusion

We studied a stochastic variant of the patient admission scheduling problem (SPAS), which aims to assign patients to rooms during their fixed LOS while considering the uncertainty of the overstay days. We consider the SPAS problem as a two-stage stochastic programming problem where the first stage assigns patients to rooms during their fixed LOS, and the second stage evaluates the expected costs resulting from patient overstay. We first proposed a scenario-based model $SPAS_{SB}$, which evaluates the expected cost by enumerating all possible scenarios. However, it is difficult to produce a solution as the model size grows exponentially with the number of scenarios. To address this difficulty, we adapt the state-variable (SV) modeling approach (Doulabi et al., 2022) to formulate the $SPAS_{SV}$ model, which is derived by reformulating the second stage of the $SPAS_{SB}$ model through the introduction of state variables, state transition constraints, and linking constraints. To solve the $SPAS_{SV}$ model efficiently, we elaborated a solution method SAA-SV in which we solve the $SPAS_{SAA}$ model to generate a high-quality feasible solution and then use it as an initial solution to solve the $SPAS_{SV}$ model. Furthermore, the SAA-SV method is embedded into a RHP framework to solve the SPAS problem with dynamic patient arrivals.

We conducted extensive computational experiments to evaluate the performance of the proposed models and methods. First, the SAA-SV method can effectively improve the solution quality and computational time compared to the direct solution of the $SPAS_{SB}$ model and the $SPAS_{SV}$ model. Second, we measured the effect of stochasticity on the SPAS problem by comparing the solutions obtained by solving the EVP and the SAA-SV method. The results demonstrate that, especially when a large penalty is applied to the extra bed usage, the implementation of the SAA-SV method yields solutions that are significantly better than those obtained by the EVP method. Third, we confirmed the contribution of the $SPAS_{SAA}$ and $SPAS_{SV}$ models in the SAA-SV method. Finally, we conducted a case study on real-life data to demonstrate the effectiveness of the RHP approach. The results show that the RHP approach can effectively support decision-making in practice. Some managerial insights can be drawn from our results, indicating that hospital administrators should strike a careful balance between patient transfers and extra bed usage to mitigate bed shortages, while aligning with resource limitations and operational priorities.

For future research, several directions could be explored. First, our proposed SAA-

SV method combines the SAA approach with state-variable modeling and provides a solution paradigm for a class of two-stage stochastic programming problems in which, for a given first-stage solution, the recourse decisions can be determined sequentially, as required by Doulabi et al. (2022). Under this problem type, both SAA and SV formulations can be proposed, thus can be solved using the SAA-SV method. Therefore, problems such as operating room scheduling (Hashemi Doulabi and Khalilpourazari, 2023), for which both SAA and SV formulations are available, are natural candidates for the proposed method. Despite its broad applicability, the challenge arises when we solve the state-variable models using general-purpose solvers like CPLEX or Gurobi, especially as the size of the instances increases. To enhance computational scalability, a customized decomposition-based framework is needed. By exploiting the specific structure of the state-variable model, this approach can integrate Benders decomposition with parallel computing to handle larger instances. Moreover, while the current framework effectively addresses discrete decision processes, a valuable extension would involve incorporating continuous-time dynamics.

Acknowledgements

We are grateful to the reviewers for their insightful comments, which helped us improve the paper significantly.

References

Barz, C. and Rajaram, K. (2015). Elective patient admission and scheduling under multiple resource constraints. *Production and Operations Management*, 24(12):1907–1930.

Bastos, L. S., Marchesi, J. F., Hamacher, S., and Fleck, J. L. (2019). A mixed integer programming approach to the patient admission scheduling problem. *European Journal of Operational Research*, 273(3):831–840.

Brandt, T., Büsing, C., and Engelhardt, F. (2025a). Patient-to-room assignment with single-rooms entitlements: Combinatorial insights and integer programming formulations. *European Journal of Operational Research*, page S0377221725001389.

Brandt, T., Klein, T. L., Reuter-Oppermann, M., Schäfer, F., Thielen, C., De Vrugt, M. V., and Viana, J. (2025b). Integrated patient-to-room and nurse-to-patient assignment in hospital wards. *OR Spectrum*.

Ceschia, S. and Schaerf, A. (2012). Modeling and solving the dynamic patient admission scheduling problem under uncertainty. *Artificial Intelligence in Medicine*, 56(3):199–205.

Ceschia, S. and Schaerf, A. (2016). Dynamic patient admission scheduling with operating room constraints, flexible horizons, and patient delays. *Journal of Scheduling*, 19(4):377–389.

Chuang, M.-T., Hu, Y.-h., and Lo, C.-L. (2018). Predicting the prolonged length of stay of general surgery patients: A supervised learning approach. *International Transactions in Operational Research*, 25(1):75–90.

Dai, J. G. and Shi, P. (2019). Inpatient overflow: An approximate dynamic programming approach. *Manufacturing & Service Operations Management*, 21(4):894–911.

De Santis, M., De Vries, S., Schmidt, M., and Winkel, L. (2022). A penalty branch-and-bound method for mixed binary linear complementarity problems. *INFORMS Journal on Computing*, 34(6):3117–3133.

Demeester, P., Souffriau, W., De Causmaecker, P., and Vanden Berghe, G. (2010). A hybrid tabu search algorithm for automatically assigning patients to beds. *Artificial Intelligence in Medicine*, 48(1):61–70.

Doulabi, H. H., Ahmed, S., and Nemhauser, G. (2022). State-variable modeling for a class of two-stage stochastic optimization problems. *INFORMS Journal on Computing*, 34(1):354–369.

Freeman, N. K., Melouk, S. H., and Mittenthal, J. (2016). A scenario-based approach for operating theater scheduling under uncertainty. *Manufacturing & Service Operations Management*, 18(2):245–261.

Gartner, D. and Kolisch, R. (2014). Scheduling the hospital-wide flow of elective patients. *European Journal of Operational Research*, 233(3):689–699.

Guido, R. (2024). Patient admission scheduling problems with uncertain length of stay: Optimization models and an efficient matheuristic approach. *International Transactions in Operational Research*, 31(1):53–87.

Hashemi Doulabi, H. and Khalilpourazari, S. (2023). Stochastic weekly operating room planning with an exponential number of scenarios. *Annals of Operations Research*, 328(1):643–664.

Izady, N. and Mohamed, I. (2021). A clustered overflow configuration of inpatient beds in hospitals. *Manufacturing & Service Operations Management*, 23(1):139–154.

Jiang, Y., Yang, F., Tang, Z., and Li, Q.-L. (2023). Admission control of hospitalization with patient gender by using markov decision process. *International Transactions in Operational Research*, 30(1):70–98.

Khatami, M., Alvarado, M., Kong, N., Parikh, P. J., and Lawley, M. A. (2022). Inpatient discharge planning under uncertainty. *IISE Transactions*, 54(4):332–347.

Ku, W.-Y. and Beck, J. C. (2016). Mixed integer programming models for job shop scheduling: A computational analysis. *Computers & Operations Research*, 73:165–173.

Liu, H., Wang, Y., and Hao, J.-K. (2024). Solving the patient admission scheduling problem using constraint aggregation. *European Journal of Operational Research*, 316(1):85–99.

Lusby, R. M., Schwierz, M., Range, T. M., and Larsen, J. (2016). An adaptive large neighborhood search procedure applied to the dynamic patient admission scheduling problem. *Artificial intelligence in medicine*, 74:21–31.

Neyshabouri, S. and Berg, B. P. (2017). Two-stage robust optimization approach to elective surgery and downstream capacity planning. *European Journal of Operational Research*, 260(1):21–40.

Ortiga, B., Salazar, A., Jovell, A., Escarrabill, J., Marca, G., and Corbella, X. (2012). Standardizing admission and discharge processes to improve patient flow: A cross sectional study. *BMC Health Services Research*, 12(1):180.

Porras, Á., Domínguez, C., Morales, J. M., and Pineda, S. (2023). Tight and compact sample average approximation for joint chance-constrained problems with applications to optimal power flow. *INFORMS Journal on Computing*, 35(6):1454–1469.

Shehadeh, K. S., Tsang, M. Y., Padman, R., and Kilic, A. (2026). Operating room-to-downstream elective surgery planning under uncertainty. *European Journal of Operational Research*, 329(1):288–307.

Turgeman, L., May, J. H., and Sciulli, R. (2017). Insights from a machine learning model for predicting the hospital length of stay (los) at the time of admission. *Expert Systems with Applications*, 78:376–385.

Vancroonenburg, W., De Causmaecker, P., and Vanden Berghe, G. (2016). A study of decision support models for online patient-to-room assignment planning. *Annals of Operations Research*, 239(1):253–271.

Vancroonenburg, W., De Causmaecker, P., and Vanden Berghe, G. (2019). Chance-constrained admission scheduling of elective surgical patients in a dynamic, uncertain setting. *Operations Research for Health Care*, 22:100196.

Wang, L. and Demeulemeester, E. (2023). Simulation optimization in healthcare resource planning: A literature review. *IISE Transactions*, 55(10):985–1007.

Wang, Y., Liu, H., Peng, B., Wang, H., and Punnen, A. P. (2023). A three-phase matheuristic algorithm for the multi-day task assignment problem. *Computers & Operations Research*, 159:106313.

Yang, C.-S., Wei, C.-P., Yuan, C.-C., and Schoung, J.-Y. (2010). Predicting the length of hospital stay of burn patients: Comparisons of prediction accuracy among different clinical stages. *Decision Support Systems*, 50(1):325–335.

Yang, X., Liu, H., Wang, Y., Punnen, A. P., and Wang, H. (2025). Integrating patient satisfaction into operating room scheduling decisions. *Computers & Industrial Engineering*, 210:111490.

Zhu, Y.-H., Toffolo, T. A. M., Vancroonenburg, W., and Vanden Berghe, G. (2019). Compatibility of short and long term objectives for dynamic patient admission scheduling. *Computers & Operations Research*, 104:98–112.

Appendix

A Mathematical models

The following subsections present the mathematical formulations of $SPAS_{SAA}$ with its dynamic variants, and dynamic variants of $SPAS_{SB}$ and $SPAS_{SV}$.

A.1 $SPAS_{SAA}$ model

The $SPAS_{SAA}$ model can be derived by replacing all scenarios with sampled scenarios Ω^R and substituting $\Pr(\omega)$ in the objective function (1) with the sampling probability. However, the search space of the $SPAS_{SAA}$ model is large due to the patient-room-admissionDay-hospitalizationDay assignment variables. By prohibiting patient transfer during their stay, we limit the search space by considering the patient-room-admissionDay assignment variables, resulting in a special case of the $SPAS_{SAA}$ model. Thus, to accelerate the solution procedure, we solve the $SPAS_{SAA}$ model without the transfers constraint.

The $SPAS_{SAA}$ model is inherited from the $SPAS_{SB}$ model by removing the variables α_{pi} and t_{pd} and replacing x_{prid} by x_{pri}, a binary variable taking the value of 1 if patient p is admitted on day i and allocated to room r, and 0 otherwise. The objective function and constraints are also adjusted accordingly. The $SPAS_{SAA}$ model is formulated as follows.

$$\begin{aligned}
\boldsymbol{SPAS}_{SAA}: \text{Min} \quad & Cost_{SAA} = \sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}_p}\sum_{i\in\mathcal{D}_p^A}\left[W^{De}(i-D_p^{EA})+L_pC_{pr}\right]x_{pri} \\
& \quad + \frac{1}{|\Omega^R|}\sum_{\omega\in\Omega^R} Q_1'(\boldsymbol{x},\omega) && (35)\\
\text{s.t.} \quad & \sum_{r\in\mathcal{R}_p}\sum_{i\in\mathcal{D}_p^A} x_{pri} = 1 \quad \forall p\in\mathcal{P} && (36)\\
& \sum_{p\in\mathcal{P}|r\in\mathcal{R}_p}\sum_{i\in\mathcal{D}_p^A|d\in\mathcal{D}_{pi}^L} x_{pri} \le B_r \quad \forall r\in\mathcal{R}, d\in\mathcal{D} && (37)\\
& x_{pri}\in\{0,1\} \quad \forall p\in\mathcal{P}, r\in\mathcal{R}_p, i\in\mathcal{D}_p^A && (38)
\end{aligned}$$

where

$$\begin{aligned}
Q_1'(\boldsymbol{x},\omega) = \text{Min} & \sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}_p}\sum_{i\in\mathcal{D}_p^A}\widetilde{L}_{p\omega}C_{pr}x_{pri} + \sum_{r\in\mathcal{R}^D}\sum_{d\in\mathcal{D}}W^{RG}b_{rd\omega} + \sum_{r\in\mathcal{R}}\sum_{d\in\mathcal{D}}W^{OP}z_{rd\omega} && (39)\\
\text{s.t.} \quad & \text{Constraints (14)} - \text{(16)}, \text{ where } \Omega^R \text{ is used instead of } \Omega
\end{aligned}$$

$$z_{rd\omega} \geq \sum_{p\in\mathcal{P}|r\in\mathcal{R}_p} \sum_{i\in\mathcal{D}_p^A|d\in\mathcal{D}_{pi}^L\cup\widetilde{\mathcal{D}}_{pi\omega}} x_{pri} - B_r \quad \forall r\in\mathcal{R}, d\in\mathcal{D}, \omega\in\Omega^R \tag{40}$$

$$\lambda_{rd}^F(u_{rd\omega}+b_{rd\omega}) \geq \sum_{p\in\mathcal{P}^F|r\in\mathcal{R}_p} \sum_{\substack{i\in\mathcal{D}_p^A|\\ d\in\mathcal{D}_{pi}^L\cup\widetilde{\mathcal{D}}_{pi\omega}}} x_{pri} \quad \forall r\in\mathcal{R}^D, d\in\mathcal{D}, \omega\in\Omega^R \tag{41}$$

$$\lambda_{rd}^M(1-u_{rd\omega}+b_{rd\omega}) \geq \sum_{\substack{p\in\mathcal{P}^M|\\ r\in\mathcal{R}_p}} \sum_{\substack{i\in\mathcal{D}_p^A|\\ d\in\mathcal{D}_{pi}^L\cup\widetilde{\mathcal{D}}_{pi\omega}}} x_{pri} \quad \forall r\in\mathcal{R}^D, d\in\mathcal{D}, \omega\in\Omega^R \tag{42}$$

A.2 Dynamic $SPAS_{SB}$ model

The key to achieving RHP&SAA-SV is to link the static version of $SPAS_{SV}$ and $SPAS_{SAA}$ models to their dynamic counterparts. On each day d^{now}, patients, who can to be scheduled, are categorized into three sets based on their admission status: $\mathcal{P}^0$-registered but not yet admitted, $\mathcal{P}^1$ - admitted and currently in the fixed LOS period, and $\mathcal{P}^2$ - admitted and currently in the overstay period. For each patient $p\in\mathcal{P}^1\cup\mathcal{P}^2$, we record the actual admission day $\bar{d}_p^A$, and update the related sets as $\mathcal{D}_p^A=\{0\}$, $\mathcal{D}_{p,0}^L=\{0,...,L_p+\bar{d}_p^A-d^{now}\}$, $\widetilde{\mathcal{D}}_{p,0,\omega}=\{1,...,L_p+\bar{d}_p^A+\widetilde{L}_{p\omega}-d^{now}\}$, and $\hat{\mathcal{D}}_{p,0}=\{d|d=1,...,\bar{d}_p^A+L_p+\max_{\omega\in\Omega}\widetilde{L}_{p\omega}-d^{now}\}$. Some new patients may not be admitted within the given planning horizon, either because their allowable admission dates are beyond the horizon or due to insufficient hospital resources. Thus, the dynamic models permit patients ($p\in\mathcal{P}^0$) to remain unadmitted in the given planning horizon. To discourage such delays and prioritize timely admissions, a high delay cost $W^{De'}=10000$ is assigned to patients who are not admitted within the planning horizon. The dynamic $SPAS_{SV}^{(d)}$ and $SPAS_{SAA}^{(d)}$ models are presented in the following, with the notation summarized in Table 10.

$SPAS_{SB}^{(d)}$ model is adapted from the $SPAS_{SB}$ model, with the following modifications:

$$\textbf{\textit{SPAS}}_{\textbf{\textit{SB}}}^{(d)}\text{: Min } Cost_{SB}|_{\mathcal{P}\to\mathcal{P}^0\cup\mathcal{P}^1} + \sum_{p\in\mathcal{P}^0} W^{De'}(|D|-D_p^{EA})(1-\sum_{i\in\mathcal{D}_p^A}\alpha_{pi}) + \sum_{p\in\mathcal{P}^1} W^{Tr}t_p^0 \tag{43}$$

s.t. Constraints (3)$|_{\mathcal{P}\to\mathcal{P}^0}$

Constraints (4) − (9)$|_{\mathcal{P}\to\mathcal{P}^0\cup\mathcal{P}^1}$

$$\sum_{i\in\mathcal{D}_p^A}\alpha_{pi}\leq 1 \quad \forall p\in\mathcal{P}^0 \tag{44}$$

$$\sum_{r\in\mathcal{R}_p} x_{pr,0,d}=1 \quad \forall p\in\mathcal{P}^1, d\in\mathcal{D}_{p,0}^L \tag{45}$$

Table 10 Notation used for the $SPAS_{SB}^{(d)}$, $SPAS_{SV}^{(d)}$ and $SPAS_{SAA}^{(d)}$ models.

Symbol	Description
Sets	
$\mathcal{P}^0 \subset \mathcal{P}$	Set of patients who have registered but not yet been admitted
$\mathcal{P}^1 \subset \mathcal{P}$	Set of patients who have been admitted and during his/her fixed LOS
$\mathcal{P}^2 \subset \mathcal{P}$	Set of patients who have been admitted and during his/her over stay period
Parameters	
$\bar{R}_p$	The room which patient p stayed on the previous day $d^{now}-1$
d^{now}	The day on which the scheduling takes place
d_p^R	The day when patient p becomes known to the system
d_p^A	The actual admission day of patient p
$W^{De'}$	Delay cost for patients not admitted within the planning horizon
First-stage decision variables	
t_p^0	1 if patient $p \in \mathcal{P}^1$ transfers on the first day of the planning horizon, 0 otherwise

$$t_p^0 \geq 1 - x_{p,\bar{R}_p,0,1} \quad \forall p \in \mathcal{P}^1 \tag{46}$$

$$t_p^0 \in \{0,1\} \quad \forall p \in \mathcal{P}^1 \tag{47}$$

Let $Q_1^{(d)}(x)$ denote the second-stage recourse function of the $SPAS_{SB}^{(d)}$ model, adapted from $Q_1(x)$ in the $SPAS_{SB}$ model. The $Q_1^{(d)}(x)$ can be expressed as follows:

$$Q_1^{(d)}(\boldsymbol{x},\omega) = \text{Min} \;\; Q_1(x)|_{\mathcal{P}\to\mathcal{P}^0\cup\mathcal{P}^1} + \sum_{p\in\mathcal{P}^2}\sum_{i\in\mathcal{D}_p^A}\sum_{d\in\widetilde{\mathcal{D}}_{pi\omega}} C_{p\bar{R}_p} \tag{48}$$

$$\text{s.t.} \qquad \text{Constraints (14)} - \text{(16)}$$

$$z_{rd\omega} \geq \sum_{p\in\mathcal{P}|r\in\mathcal{R}_p}\sum_{i\in\mathcal{D}_p^A}\Big[\mathbb{1}(d\in\mathcal{D}_{pi}^L)x_{prid} + \mathbb{1}(d\in\widetilde{\mathcal{D}}_{pi\omega}, p\in\mathcal{P}^0\cup\mathcal{P}^1)x_{pri,i+L_p-1} + \mathbb{1}(d\in\widetilde{\mathcal{D}}_{pi\omega}, p\in\mathcal{P}^2, r=\bar{R}_p)\Big] - B_r \quad \forall r\in\mathcal{R}, d\in\mathcal{D}, \omega\in\Omega \tag{49}$$

$$\lambda_{rd}^F(u_{rd\omega}+b_{rd\omega}) \geq \sum_{p\in\mathcal{P}^F|r\in\mathcal{R}_p}\sum_{i\in\mathcal{D}_p^A}\big[\mathbb{1}(d\in\mathcal{D}_{pi}^L)x_{prid} + \mathbb{1}(d\in\widetilde{\mathcal{D}}_{pi\omega}, p\in\mathcal{P}^0\cup\mathcal{P}^1)x_{pri,i+L_p-1} + \mathbb{1}(d\in\widetilde{\mathcal{D}}_{pi\omega}, p\in\mathcal{P}^2, r=\bar{R}_p)\Big] \quad \forall r\in\mathcal{R}^D, d\in\mathcal{D}, \omega\in\Omega \tag{50}$$

$$\lambda_{rd}^M(1-u_{rd\omega}+b_{rd\omega}) \geq \sum_{p\in\mathcal{P}^M|r\in\mathcal{R}_p}\sum_{i\in\mathcal{D}_p^A}\big[\mathbb{1}(d\in\mathcal{D}_{pi}^L)x_{prid} + \mathbb{1}(d\in\widetilde{\mathcal{D}}_{pi\omega}, p\in\mathcal{P}^0\cup\mathcal{P}^1)x_{pri,i+L_p-1} + \mathbb{1}(d\in\widetilde{\mathcal{D}}_{pi\omega}, p\in\mathcal{P}^2, r=\bar{R}_p)\Big] \quad \forall r\in\mathcal{R}^D, d\in\mathcal{D}, \omega\in\Omega \tag{51}$$

In the first-stage, the objective function (43) builds upon that of the $SPAS_{SB}$ model by incorporating the delay cost for patients who are not admitted within the planning horizon,

as well as the transfer cost for patients ($p \in \mathcal{P}1$) who are transferred on the first day of the planning horizon. Constraint (44) ensures that each patient can be admitted only once and within the range of admission date. Constraint (45) ensures that patient $p \in \mathcal{P}^1$ must be assigned to a room for every day of their hospitalization. Constraint (46) ensures that the auxiliary variable t_p^0 becomes 1 if patient p transfers on the first day of the planning horizon. Constraint (47) defines the domain of the t_p^0. The second-stage objective function (48) is similar to that of the $Q_1(x)$ in the $SPAS_{SB}$ model. Constraints (49) - (51) are the same as constraints (11) - (13).

A.3 Dynamic $SPAS_{SV}$ model

Our $SPAS_{SV}^{(d)}$ model is adapted from the $SPAS_{SV}$ model, following the same modifications applied in $SPAS_{SB} \rightarrow SPAS_{SB}^{(d)}$.

$$\boldsymbol{SPAS_{SV}^{(d)}}\text{: Min } Cost_{SV}|_{\mathcal{P}\rightarrow\mathcal{P}^0\cup\mathcal{P}^1} + \sum_{p\in\mathcal{P}^0} W^{De'}(|D| - D_p^{EA})(1 - \sum_{i\in\mathcal{D}_p^A}\alpha_{pi}) + \sum_{p\in\mathcal{P}^1} W^{Tr}t_p^0 \tag{52}$$

$$\text{s.t.} \quad \text{Constraints (3)}|_{\mathcal{P}\rightarrow\mathcal{P}^0}$$
$$\text{Constraints (4)} - \text{(9)}|_{\mathcal{P}\rightarrow\mathcal{P}^0\cup\mathcal{P}^1}$$
$$\text{Constraints (44)} - \text{(47)}$$

Let $Q_2^{(d)}(x)$ denote the second-stage recourse function of the $SPAS_{SV}^{(d)}$ model, adapted from $Q_2(x)$ in the $SPAS_{SV}$ model. The $Q_2^{(d)}(x)$ can be expressed as follows:

$$Q_2^{(d)}(\boldsymbol{x}) = \text{Min } \; Q_2(x)|_{\mathcal{P}\rightarrow\mathcal{P}^0\cup\mathcal{P}^1} + \sum_{p\in\mathcal{P}^2}\sum_{d\in\hat{\mathcal{D}}_{p,0}} \Pr(p, d - \bar{d}_p^A + 1)C_{p,\bar{R}_p} \tag{53}$$

$$\text{s.t.} \quad \text{Constraints (25), (26), (31)} - \text{(34)}$$
$$\sum_{n\in\mathcal{N}'_{\tau dr}} y^1_{dr\tau ni} = \mathbb{1}\left(d \in \mathcal{D}^L_{p_{\tau+1}i}\right) x_{p_{\tau+1}rid}$$
$$+\mathbb{1}\left(d \in \hat{\mathcal{D}}_{p_{\tau+1}i}, p_{\tau+1} \in \mathcal{P}^0 \cup \mathcal{P}^1\right) x_{p_{\tau+1}ri,i+L_{p_{\tau+1}}-1} + \mathbb{1}(d \in \hat{\mathcal{D}}_{p_{\tau+1}i}, p_{\tau+1} \in \mathcal{P}^2, r = \bar{R}_p)$$
$$\forall d \in \mathcal{D}, r \in \mathcal{R}, \tau \in \mathcal{T}_{dr} \setminus \{|\hat{\mathcal{P}}_{dr}|\}, i \in \mathcal{D}^A_{p_{\tau+1}} \tag{54}$$
$$\sum_{n\in\mathcal{N}_{\tau dr}} y^0_{dr\tau n} = 1 - \sum_{i\in\mathcal{D}^A_{p_{\tau+1}}} \Big[\mathbb{1}\left(d \in \mathcal{D}^L_{p_{\tau+1}i}\right) x_{p_{\tau+1}rid}$$
$$+\mathbb{1}\left(d \in \hat{\mathcal{D}}_{p_{\tau+1}i}, p_{\tau+1} \in \mathcal{P}^0 \cup \mathcal{P}^1\right) x_{p_{\tau+1}ri,i+L_{p_{\tau+1}}-1} + \mathbb{1}(d \in \hat{\mathcal{D}}_{p_{\tau+1}i}, p_{\tau+1} \in \mathcal{P}^2, r = \bar{R}_p)\Big]$$
$$\forall d \in \mathcal{D}, r \in \mathcal{R}, \tau \in \mathcal{T}_{dr} \setminus \{|\hat{\mathcal{P}}_{dr}|\} \tag{55}$$

$$\sum_{g\in\mathcal{G}_{\tau dr}} q^1_{dr\tau gi} = \mathbb{1}\left(d\in\mathcal{D}^L_{p_{\tau+1}i}\right) x_{p_{\tau+1}rid}$$
$$+\mathbb{1}\left(d\in\hat{\mathcal{D}}_{p_{\tau+1}i}, p_{\tau+1}\in\mathcal{P}^0\cup\mathcal{P}^1\right) x_{p_{\tau+1}ri,i+L_{p_{\tau+1}}-1} + \mathbb{1}(d\in\hat{\mathcal{D}}_{p_{\tau+1}i}, p_{\tau+1}\in\mathcal{P}^2, r=\bar{R}_p)$$
$$\forall d\in\mathcal{D}, r\in\mathcal{R}^D, \tau\in\mathcal{T}_{dr}\setminus\{|\hat{\mathcal{P}}_{dr}|\}, i\in\mathcal{D}^A_{p_{\tau+1}} \tag{56}$$
$$\sum_{g\in\mathcal{G}_{\tau dr}} q^0_{dr\tau g} = 1 - \sum_{i\in\mathcal{D}^A_{p_{\tau+1}}} \Big[\mathbb{1}\left(d\in\mathcal{D}^L_{p_{\tau+1}i}\right) x_{p_{\tau+1}rid}$$
$$+\mathbb{1}\left(d\in\hat{\mathcal{D}}_{p_{\tau+1}i}, p_{\tau+1}\in\mathcal{P}^0\cup\mathcal{P}^1\right) x_{p_{\tau+1}ri,i+L_{p_{\tau+1}}-1} + \mathbb{1}(d\in\hat{\mathcal{D}}_{p_{\tau+1}i}, p_{\tau+1}\in\mathcal{P}^2, r=\bar{R}_p)\Big]$$
$$\forall d\in\mathcal{D}, r\in\mathcal{R}^D, \tau\in\mathcal{T}_{dr}\setminus\{|\hat{\mathcal{P}}_{dr}|\} \tag{57}$$

A.4 Dynamic $SPAS_{SAA}$ model

Our $SPAS^{(d)}_{SAA}$ model is adapted from the $SPAS_{SAA}$ model, following the same modifications applied in $SPAS_{SB} \rightarrow SPAS^{(d)}_{SB}$.

$$\boldsymbol{SPAS}^{(d)}_{SAA}: \quad \text{Min } Cost_{SAA}|_{\mathcal{P}\rightarrow\mathcal{P}^0\cup\mathcal{P}^1} + \sum_{p\in\mathcal{P}^1} W^{Tr} t^0_p$$
$$+ \sum_{p\in\mathcal{P}^0}\left[W^{De'}(|D| - D^{EA}_p)\right](1 - \sum_{r\in\mathcal{R}_p}\sum_{i\in\mathcal{D}^A_p} x_{pri}) \tag{58}$$

s.t. Constraints (37), (38)$|_{\mathcal{P}\rightarrow\mathcal{P}^0\cup\mathcal{P}^1}$

$$\sum_{r\in\mathcal{R}_p}\sum_{i\in\mathcal{D}^A_p} x_{pri} \le 1 \quad \forall p\in\mathcal{P}^0 \tag{59}$$
$$\sum_{r\in\mathcal{R}_p} x_{pr,1} = 1 \quad \forall p\in\mathcal{P}^1 \tag{60}$$
$$t^0_p \ge 1 - x_{pri}, \quad \forall p\in\mathcal{P}^1, r=\bar{R}_p, i\in\mathcal{D}^A_p \tag{61}$$
$$t^0_p \in \{0,1\} \quad \forall p\in\mathcal{P}^1 \tag{62}$$

Let $Q'^{(d)}_1(x,\omega)$ denote the second-stage recourse function of the $SPAS^{(d)}_{SAA}$ model, adapted from $Q'_1(x,\omega)$ in the $SPAS_{SAA}$ model. The $Q'^{(d)}_1(x,\omega)$ can be expressed as follows:

$$Q'^{(d)}_1(\boldsymbol{x},\omega) = \text{Min } Q'_1(\boldsymbol{x},\omega)|_{\mathcal{P}\rightarrow\mathcal{P}^0\cup\mathcal{P}^1} + \sum_{p\in\mathcal{P}^2}\sum_{i\in\mathcal{D}^A_p} \widetilde{L}_{p\omega} C_{p\bar{R}_p} \tag{63}$$

s.t. Constraints (14) – (16), where Ω^R is used instead of Ω

$$z_{rd\omega} \ge \sum_{p\in\mathcal{P}^0\cup\mathcal{P}^1|r\in\mathcal{R}_p}\ \sum_{i\in\mathcal{D}^A_p|d\in\mathcal{D}^L_{pi}\cup\widetilde{\mathcal{D}}_{pi\omega}} x_{pri} + \mathbb{1}(d\in\widetilde{\mathcal{D}}_{pi\omega}, p\in\mathcal{P}^2, r=\bar{R}_p) - B_r$$
$$\forall r\in\mathcal{R}, d\in\mathcal{D}, \omega\in\Omega^R \tag{64}$$

$$\lambda^F_{rd}(u_{rd\omega} + b_{rd\omega}) \geq \sum_{p\in\mathcal{P}^F\cap(\mathcal{P}^0\cup\mathcal{P}^1)|r\in\mathcal{R}_p} \sum_{i\in\mathcal{D}^A_p|d\in\mathcal{D}^L_{pi}\cup\widetilde{\mathcal{D}}_{pi\omega}} x_{pri}$$
$$+\mathbb{1}(d \in \widetilde{\mathcal{D}}_{pi\omega}, p \in \mathcal{P}^F \cap \mathcal{P}^2, r = \bar{R}_p) \quad \forall r \in \mathcal{R}^D, d \in \mathcal{D}, \omega \in \Omega^R \tag{65}$$
$$\lambda^M_{rd}(1 - u_{rd\omega} + b_{rd\omega}) \geq \sum_{p\in\mathcal{P}^M\cap(\mathcal{P}^0\cup\mathcal{P}^1)|r\in\mathcal{R}_p} \sum_{i\in\mathcal{D}^A_p|d\in\mathcal{D}^L_{pi}\cup\widetilde{\mathcal{D}}_{pi\omega}} x_{pri}$$
$$+\mathbb{1}(d \in \widetilde{\mathcal{D}}_{pi\omega}, p \in \mathcal{P}^M \cap \mathcal{P}^2, r = \bar{R}_p) \quad \forall r \in \mathcal{R}^D, d \in \mathcal{D}, \omega \in \Omega^R \tag{66}$$

A.5 Impact of patient sequencing on the state-variable model $SPAS_{SV}$

The choice of patient sequence $\hat{\mathcal{P}}_{dr}$ can indeed influence the computational performance of the SV model. Specifically, the ordering of patients can affect the structure of the gender occupancy state space $G_{\tau rd} = \{\text{None, OnlyM, OnlyF, Mixed}\}$, corresponding to rooms occupied by no patients, male patients only, female patients only, or by both genders, respectively. This structure directly determines the number of state variables $q^1_{rd\tau gi}$, $\forall d \in \mathcal{D}, r \in \mathcal{R}^D, \tau \in \mathcal{T}_{rd} \setminus \{|\hat{\mathcal{P}}^0_{rd}|\}, i \in \mathcal{D}^A_{p_{\tau+1}}, g \in \mathcal{G}_{\tau rd}$, and the number of state variables $q^0_{rd\tau g}, \forall d \in \mathcal{D}, r \in \mathcal{R}^D, \tau \in \mathcal{T}_{rd}, g \in \mathcal{G}_{\tau rd}$ in the state variable model $SPAS_{SV}$. For example, consider an example with 1 room with 4 beds and follows policy D ($\mathcal{R}^D = \{1\}$), one-day planning horizon ($\mathcal{D} = \{1\}$), and 2 male (M) patients and 1 female (F) patient ($\hat{\mathcal{P}}_{1,1} = \{p_1, p_2, p_3\}$). Each patient can be admitted on day 1 ($\mathcal{D}^A_p = \{1\}$, $\forall p \in \hat{\mathcal{P}}_{1,1}$), and the LOS of each patient is 1 day. Thus, we define $\tau = 0, 1, 2, 3$ as the index for the patient sequence, where $\tau = 0$ corresponds to the initial state before any patient is admitted, and $\tau = 1, 2, 3$ correspond to the states after admitting the first, second, and third patient in the sequence, respectively. The possible patient sequences and its corresponding room gender occupancy state could be:

- Sequence $[M, M, F]$ leads to states $G_{0rd} = \{None\}$, $G_{1rd} = \{None, OnlyM\}$, $G_{2rd} = \{None, OnlyM\}$, $G_{3rd} = \{None, OnlyM, OnlyF, Mixed\}$. The number of $q^1_{rd\tau gi}$ variables is 5 (where d, r, i can only take value 1, and $\tau \in \mathcal{T}_{rd} \setminus \{|\hat{\mathcal{P}}^0_{rd}|\}$ can take value 0, 1, 2, and g can take values corresponding to the states in $G_{\tau rd}$). The number of $q^0_{rd\tau g}$ variables is 9 (where d, r can only take value 1, $\tau \in \mathcal{T}_{rd}$ can take value 0, 1, 2, 3, and g can take values corresponding to the states in $G_{\tau rd}$). Thus, the total number of $q^1_{rd\tau gi}$ and $q^0_{rd\tau g}$ variables is 14.

- Sequence $[F, M, M]$ leads to states $G_{0rd} = \{None\}$, $G_{1rd} = \{None, OnlyF\}$, $G_{2rd} =$

$\{None, OnlyF, OnlyM, Mixed\}$, $G_{3rd} = \{None, OnlyM, OnlyF, Mixed\}$. The total number of $q^1_{rd\tau gi}$ and $q^0_{rd\tau g}$ variables is 18 (the number of $q^1_{rd\tau gi}$ variables is 7, and $q^0_{rd\tau g}$ variables is 11).

Thus, different patient sequences can lead to different state spaces and numbers of state variables, which in turn can affect the size and complexity of the SV model.

B SAA-SV Method

B.1 Main scheme of the SAA-SV method

Algorithm 1 outlines the main scheme of the SAA-SV method for solving the SPAS problem. The SAA-SV method starts by generating a set of scenarios Ω^R and initializing the necessary parameters. After that, it solves the $SPAS_{SAA}$ model within a time limit TL_{SAA} to obtain an approximate solution $\boldsymbol{x}'$. The complete solution ($\boldsymbol{\alpha}$, $\boldsymbol{x}$, $\boldsymbol{t}$), which is the solution corresponding to $\boldsymbol{x}'$ in $SPAS_{SV}$ model, is obtained as follows: Set $\alpha_{pi} = \sum_{r \in \mathcal{R}_p} x_{pri}$ for all $p \in \mathcal{P}, i \in \mathcal{D}^A_p$; Set $x_{prid} = x_{pri}$ for all $p \in \mathcal{P}, r \in \mathcal{R}_p, i \in \mathcal{D}^A_p, d \in \mathcal{D}^L_{pi}$; Set $t_{pd} = 1$ for all $p \in \mathcal{P} | L_p \geq 2, d = D^{EA}_p + 1, ..., D^{LA}_p + L_p - 1$. The objective value Obj of the $\boldsymbol{x}'$ is obtained by fixing the first-stage variables based on solution ($\boldsymbol{\alpha}$, $\boldsymbol{x}$, $\boldsymbol{t}$) in the $SPAS_{SV}$ model and solving the model. The best solution $(\boldsymbol{\alpha}_{SAA}, \boldsymbol{x_{SAA}}, \boldsymbol{t_{SAA}})$ along with its objective value Obj_{SAA} are updated if a better solution is found. We utilize the remaining time TL_{SAA} - $elapsed_time$ to search for new solutions using a different set of scenarios Ω^R and launch a new round SAA procedure. Moreover, we set a maximum number of iterations $MaxIter$ for the SAA procedure. The above procedure is repeated until either the time limit TL_{SAA} or the maximum number of iterations $MaxIter$ is reached. Then, the $SPAS_{SV}$ model is solved using $(\boldsymbol{\alpha}_{SAA}, \boldsymbol{x_{SAA}}, \boldsymbol{t_{SAA}})$ as the initial solution within the remaining time.

B.2 Scenarios generation

The performance of the SAA procedure is dependent on the number of samples available (Porras et al., 2023). Generally, more samples lead to a better approximation of the objective function, but also increase the computational efforts. Thus, scenario selection is crucial for obtaining a high-quality solution of the $SPAS_{SAA}$ model. We are inspired from the approach of Freeman et al. (2016) to use a relatively small number of scenarios to

Algorithm 1: Outline of the SAA-SV method for the SPAS problem

Input : A problem instance NP, the time limit TL_{SAA} and TTL, the maximum number $MaxIter$ of SAA iterations, the number of scenarios $|\Omega^R|$.

Output: The best solution $(\boldsymbol{\alpha}^*, \boldsymbol{x}^*, \boldsymbol{t}^*)$ with objective value Obj^* found so far.

Initialize the best objective value in SAA process $Obj_{SAA} \leftarrow +\infty$, set the iteration counter $iter \leftarrow 0$, and the time limit for the current iteration $TL^{(iter)} \leftarrow TL_{SAA}$;

while *iter* $< MaxIter$ *and the elapsed time does not reach* TL_{SAA} **do**

  $\Omega^R \leftarrow$ ScenarioGeneration($NP, |\Omega^R|$);

  if $iter = 0$ **then**

    $\boldsymbol{x}' \leftarrow$ SolveModel($SPAS_{SAA}$, NP, Ω^R, $TL^{(iter)}$);

  else

    $\boldsymbol{x}' \leftarrow$ SolveModelWS($SPAS_{SAA}$, NP, Ω^R, $\boldsymbol{x'_{SAA}}$, $TL^{(iter)}$);

  $(\boldsymbol{\alpha}, \boldsymbol{x}, \boldsymbol{t}) \leftarrow$ ChangeSolution($\boldsymbol{x}'$);

  $Obj \leftarrow$ Fix-and-Solve($SPAS_{SV}$, NP, $\boldsymbol{\alpha}$, $\boldsymbol{x}$, $\boldsymbol{t}$);

  if $Obj < Obj_{SAA}$ **then**

    $(\boldsymbol{\alpha}_{SAA}, \boldsymbol{x_{SAA}}, \boldsymbol{t_{SAA}}) \leftarrow (\boldsymbol{\alpha}, \boldsymbol{x}, \boldsymbol{t})$, $\boldsymbol{x'_{SAA}} \leftarrow \boldsymbol{x}'$, $Obj_{SAA} \leftarrow Obj$;

  $iter \leftarrow iter + 1$, $TL^{(iter)} \leftarrow TL_{SAA} -$ elapsed_time;

$(\boldsymbol{\alpha}^*, \boldsymbol{x}^*, \boldsymbol{t}^*, Obj^*) \leftarrow$ SolveModel($SPAS_{SV}$, NP, $\boldsymbol{\alpha}_{SAA}$, $\boldsymbol{x_{SAA}}, \boldsymbol{t_{SAA}}$, $TTL -$ elapsed_time);

return $(\boldsymbol{\alpha}^*, \boldsymbol{x}^*, \boldsymbol{t}^*, Obj^*)$;

provide information regarding the overstay distributions. In detail, we generate a certain number of scenarios $|\Omega^R|$, where the first scenario of the Ω^R uses the median-based overstay for each patient. The median-based overstay days of patient p is $\widetilde{L}_p = \arg\min\{l - L_p - 1 : \Pr(p, l) < 0.5\}$. The remaining scenarios are constructed by random sampling to capture the overstay distribution. Note that median-based overstay is utilized for each patient instead of expected overstay. The reason is that $SPAS_{SAA}$ model requires deterministic integer overstay values. In contrast, the true expected overstay for a patient is generally a fractional number. It is defined as $\mathbb{E}[overstay_p] = \sum_{k=1}^{\widehat{L}_p - L_p} k \cdot \Pr(overstay_p = k)$, where L_p is the fixed length of stay (LOS), $\widehat{L}_p$ is the maximum hospitalization duration,

and $\Pr(overstay_p = k)$ is the probability of the patient staying for exactly k days beyond L_p. Moreover, since LOS data are typically right-skewed, the true expected overstay, after rounding, may lead to a more conservative deterministic overstay value. In contrast, the median-based overstay provides an integer representative value based on the 0.5 cutoff of the survival probability curve. Furthermore, we found that the true expected overstays are close to the median-based overstays in most instances.

B.3 Comparison of model sizes between $SPAS_{SV}$ and $SPAS_{SAA}$

In this subsection, we compare the model sizes of the $SPAS_{SV}$ and $SPAS_{SAA}$ formulations to highlight the advantage of the state-variable-based formulation. To compare the model sizes, we assess the order of magnitude of the number of variables and constraints in the two formulations. Due to the complexity of the models, we rely on leading-order terms that dominate the formulation size, rather than enumerating all variables and constraints explicitly. This type of size comparison based on dominant components and their scaling behavior is standard in the operation research literature (Ku and Beck, 2016). For the $SPAS_{SV}$ model, the size of the first-stage formulation is dominated by x_{prid} variables, while the second-stage formulation is dominated by the state variables $y^1_{rd\tau ni}$, $y^0_{rd\tau n}$, $q^1_{rd\tau gi}$, and $q^0_{rd\tau g}$. For the $SPAS_{SAA}$ model, the size of the first-stage formulation is dominated by x_{pri} variables, while the second-stage formulation is dominated by scenario-specific variables $z_{rd\omega}, u_{rd\omega}$, and $b_{rd\omega}$. Based on this analysis, we summarize the number of variables and constraints of the two models in Table 11. From the table, it is evident that the key difference lies in the scaling of the second-stage formulation. In the $SPAS_{SAA}$ approach, the total number of scenarios $|\Omega^R|$ grows exponentially with the number of patients, specifically $(M+1)^N$, where N is the number of patients and $M+1$ is the number of possible overstay durations. This leads to the "curse of dimensionality" as the problem scale increases.

In contrast, the $SPAS_{SV}$ formulation collapses this exponential space into a compact room-day state space of size $\bar{T}\bar{S}(\bar{A}+1)$, where $\bar{T}$ is the average number of periods per room per day, $\bar{S}$ is the average number of room states per room per day, and $\bar{A}$ is the average size of the admission window $|\mathcal{D}^A_p|$. The factor $\bar{A}$ corresponds to the state variables indexed by admission day, such as $q^1_{rd\tau gi}$ and $y^1_{rd\tau ni}$, while the additional $(+1)$ accounts for the corresponding state variables without the admission-day index, such as $q^0_{rd\tau g}$ and

$y^0_{rd\tau n}$. All three dimensions have strict practical limits. Specifically, $\bar{T}$ is bounded by the maximum number of patients per day $\hat{P}^{max}_{rd}$, and $\bar{S}$ is determined by the room gender states ($|G_{\tau rd}| = 4$) and the maximum bed capacity ($B_r + O_r$). Although the number of possible gender states in period τ varies with the patient sequence (as detailed in Appendix A.5), the maximum number of possible gender states in any single period is strictly capped at $|G_{\tau rd}| = 4$. Consequently, the size of the state variables scales by $\bar{T}\bar{S}(\bar{A}+1)$, effectively accommodating the admission day index i. Furthermore, because the constraints naturally aggregate over the admission days in state transition constraints, the number of constraints scales even more compactly by $\bar{T}\bar{S}$. This fundamental structural advantage ensures that $SPAS_{SV}$ maintains computational tractability by avoiding the exponential explosion of scenarios inherent in the SAA formulation.

Table 11: Comparison of model sizes of $SPAS_{SV}$ and $SPAS_{SAA}$ (dominant components).

Model	Number of Variables	Number of Constraints
$SPAS_{SV}$	First stage: $\|\mathcal{P}\|\bar{R}\bar{A}\bar{L}$ + Second stage: $\|\mathcal{R}\|\|\mathcal{D}\|\bar{T}\bar{S}(\bar{A}+1)$	First stage: $\|\mathcal{P}\|\|\mathcal{D}\|\bar{R}$ + Second stage: $\|\mathcal{R}\|\|\mathcal{D}\|\bar{T}\bar{S}$
$SPAS_{SAA}$	First stage: $\|\mathcal{P}\|\bar{R}\bar{A}$ + Second stage: $\|\mathcal{R}\|\|\mathcal{D}\|\|\Omega^R\|$	First stage: $\|\mathcal{R}\|\|\mathcal{D}\|$ + Second stage: $\|\mathcal{R}\|\|\mathcal{D}\|\|\Omega^R\|$

$\bar{R}$ - average value of $|\mathcal{R}_p|$, $\bar{A}$ - average value of $|\mathcal{D}^A_p|$, $\bar{L}$ - average value of $|\mathcal{D}^L_{pi}|$,
$\bar{T}$ - average number of total periods per room per day
$\bar{S}$ - average number of total room states per room per day

Consider an instance with 100 patients, 20 rooms (all under gender policy D), and a 14-day planning horizon. Each patient can be assigned to any room on any day, with a fixed LOS of 7 days and a potential overstay of up to 2 days. Each room has a base capacity of 4 beds plus one additional extra bed. Under these settings, the average parameters are estimated as $\bar{R} = 20$, $\bar{A} = 14$, $\bar{T} = 100$, $\bar{L} = 7$, and $\bar{S} = 9$ (where $\bar{S} = 5 + 4$ accounts for the maximum bed capacity and the four possible gender states per room-day). In this case, the $SPAS_{SAA}$ model with $|\Omega^R| = 100000$ scenarios (which is a very small number compared to the total scenario $(M+1)^N = 3^{100}$) would have on the order of $100 \times 20 \times 14 + 20 \times 14 \times 100000 = 28,028,000$ variables and $20 \times 14 + 20 \times 14 \times$

$100000 = 28,000,280$ constraints. In contrast, the $SPAS_{SV}$ model would have on the order of $100 \times 20 \times 14 \times 7 + 20 \times 14 \times 100 \times 9 \times (14+1) = 3,976,000$ variables and $100 \times 20 \times 14 + 20 \times 14 \times 100 \times 9 = 280,000$ constraints. We use $|\Omega^R| = 100000$ here only to illustrate the exponential gap in the total scenario space. This illustrates how the SV-based formulation can significantly reduce model size and improve computational tractability compared to the SAA-based formulation.

C RHP & SAA-SV

RHP & SAA-SV: To implement the RHP, two key notations are introduced: *Current day* d^{now}, represents the day on which the scheduling takes place. *Registration day* d_p^R, represents the day when patient p becomes known to the system. On day d^{now}, the hospital collects new demand information, including newly registered patients and discharge records, and updates the patient set. Only patients with $d_p^R \leq d^{now}$ and who have not been discharged are considered. Thus, the SAA-SV is applied iteratively, starting with $d^{now} = 0$ and increasing d^{now} by one at each iteration. For each d^{now}, the solution from the previous iteration is recorded, and all the assignments for the day $d^{now} - 1$ are fixed. The process continues until the termination condition is met. The planning horizon is set to 14 days, in line with the hospital's typical two-week scheduling practice. The key to achieving RHP&SAA-SV is to link the static version of $SPAS_{SV}$ and $SPAS_{SAA}$ models to their dynamic counterparts, which are given in the Appendix A.

D Proof of Theorem 1

Proof. To prove the validity of the $SPAS_{SV}$ model by induction, we show that the model accurately computes the values of the variable pairs $y^1_{dr\tau ni}, y^0_{dr\tau n}$ and $q^1_{dr\tau gi}, q^0_{dr\tau g}$. Specifically, the variables $y^1_{rd\tau ni}$ and $y^0_{rd\tau n}$ describe the bed-occupancy state, while the variables $q^1_{rd\tau gi}$ and $q^0_{rd\tau g}$ describe the room-gender state. Therefore, we prove the correctness of these two groups of state variables separately.

Proof for the variable pairs $y^1_{dr\tau ni}$ ***and*** $y^0_{dr\tau n}$***:*** We first consider the case $\tau = 0$ as the basis of induction. In constraints (27) and (28), we have $\mathcal{N}_{0,dr} = \{0\}$. Therefore, we

can rewrite these two constraints for $\tau = 0$ as follows.

$$y^1_{dr,0,0,i} = \mathbb{1}\left(d \in \mathcal{D}^L_{p_1 i}\right) x_{p_1 rid} + \mathbb{1}\left(d \in \hat{\mathcal{D}}_{p_1 i}\right) x_{p_1 ri,i+L_{p_1}-1} \tag{67}$$
$$\forall d \in \mathcal{D}, r \in \mathcal{R}, i \in \mathcal{D}^A_{p_1}$$
$$y^0_{dr,0,0} = 1 - \sum_{i \in \mathcal{D}^A_{p_1}} \left[\mathbb{1}\left(d \in \mathcal{D}^L_{p_1 i}\right) x_{p_1 rid} + \mathbb{1}\left(d \in \hat{\mathcal{D}}_{p_1 i}\right) x_{p_1 ri,i+L_{p_1}-1}\right] \tag{68}$$
$$\forall d \in \mathcal{D}, r \in \mathcal{R}$$

Constraints (67) and (68) show that for $\tau = 0$, $y^1_{dr\tau ni}$ and $y^0_{dr\tau n}$ are defined properly. Suppose that variables $y^1_{dr,\tau-1,ni}$ and $y^0_{dr,\tau-1,n}$ are already computed correctly. As the inductive period, we need to show that the model computes the values of $y^1_{dr\tau ni}$ and $y^0_{dr\tau n}$ correctly.

Before discussing constraint (25), we clarify the indexing convention of the state variables. The index τ represents the state after patients $p_1, \ldots, p_\tau$ have been processed. Therefore, the transition from period $\tau - 1$ to period τ is driven by patient p_τ. More specifically, $y^1_{rd\tau ni}$ is defined as a joint state-linking variable: it represents the probability that the number of occupied beds in room r on day d equals n after patients $p_1, \ldots, p_\tau$ have been processed, and that a bed in the same room is reserved for the next patient $p_{\tau+1}$ admitted on day i. Hence, the index i in $y^1_{rd\tau ni}$ is associated with the admission-day set $\mathcal{D}^A_{p_{\tau+1}}$, whereas patient p_τ appears in the transition terms from period $\tau - 1$ to period τ. With this convention, the right-hand side of constraint (25) represents the probability that the cumulative used beds in room r on day d by patients p_1 to p_τ is equal to n. Also with respect to constraints (27) and (28), it is clear that only one of the variables $y^0_{dr\tau n}$ and $y^1_{dr\tau ni}, \forall i \in \mathcal{D}^A_{p_{\tau+1}}$ can be nonzero. Note that since constraints (2) and (3) hold, at most one of the variables $y^1_{dr\tau ni}, \forall i \in \mathcal{D}^A_{p_{\tau+1}}$ can be nonzero. Therefore, only one variable that is not equal to 0 appears on the right-hand side of constraint (25).

For a clear statement, we use $E^A_{dr\tau n}$ to represent the event that the cumulative occupied beds in room r on day d by patients p_1 to p_τ is equal to n. Recall that $\Pr(\cdot)$ denotes the probability of event $(\cdot)$. Here, $\Pr(E)$ is the probability of event E, and $\Pr(p, l)$ is the probability that patient p remains hospitalized on l-th day after admission. In the second-stage probability calculation, the first-stage decision variables are fixed. Therefore, expressions involving x_{prid} are used only to indicate deterministic conditions induced by a selected first-stage solution, rather than additional random events. When $\sum_{i \in \mathcal{D}^A_{p_\tau}} x_{p_\tau rid} =$

1, the index i in $\Pr(p_\tau, d-i+1)$ refers to the unique admission day selected for patient p_τ by the fixed first-stage solution. In the final equality, this case is equivalently represented by summing over all $i \in \mathcal{D}^A_{p_\tau}$, because at most one corresponding state-linking variable can be nonzero. To complete the proof, we need to show that $\Pr(E^A_{dr\tau n})$ is equal to left-hand side of constraint (25).

$$\begin{aligned}
\Pr(E^A_{dr\tau n}) &= \sum_{n' \in \{n-1,n\}} \Pr(E^A_{dr,\tau-1,n'}) \Pr(E^A_{dr\tau n} | E^A_{dr,\tau-1,n'}) \\
&= \sum_{n' \in \{n-1,n\}} \Pr(E^A_{dr,\tau-1,n'}) \left[\Pr\left(\sum_{i \in \mathcal{D}^A_{p_\tau}} x_{p_\tau rid} = 1 \right) \Pr\left(E^A_{dr\tau n} | E^A_{dr,\tau-1,n'} \wedge \sum_{i \in \mathcal{D}^A_{p_\tau}} x_{p_\tau rid} = 1 \right) \right. \\
&\quad \left. + \Pr\left(\sum_{i \in \mathcal{D}^A_{p_\tau}} x_{p_\tau rid} = 0 \right) \Pr\left(E^A_{dr\tau n} | E^A_{dr,\tau-1,n'} \wedge \sum_{i \in \mathcal{D}^A_{p_\tau}} x_{p_\tau rid} = 0 \right) \right]
\end{aligned}$$

It is clear that $\Pr\left(E^A_{dr\tau n} | E^A_{dr,\tau-1,n'} \wedge \sum_{i \in \mathcal{D}^A_{p_\tau}} x_{p_\tau rid} = 0\right) = 1$ holds for $n' = n$ if $n \in \mathcal{N}_{\tau-1,dr}$. For $n' = n-1$, we have $\Pr\left(E^A_{dr\tau n} | E^A_{dr,\tau-1,n'} \wedge \sum_{i \in \mathcal{D}^A_{p_\tau}} x_{p_\tau rid} = 0\right) = 0$. Also $\Pr\left(E^A_{dr\tau n} | E^A_{dr,\tau-1,n'} \wedge \sum_{i \in \mathcal{D}^A_{p_\tau}} x_{p_\tau rid} = 1\right) = \Pr(p_\tau, d-i+1)$ holds. Considering these points, we can rewrite the above relation as follows.

$$\begin{aligned}
\Pr(E^A_{dr\tau n}) =& \mathbb{1}(n \in \mathcal{N}_{\tau-1,dr}) \Pr\left(E^A_{dr,\tau-1,n}\right) \Pr\left(\sum_{i \in \mathcal{D}^A_{p_\tau}} x_{p_\tau rid} = 0 \right) \\
&+ \sum_{n' \in \{n-1,n\}} \left[\Pr\left(\sum_{i \in \mathcal{D}^A_{p_\tau}} x_{p_\tau rid} = 1 \right) \Pr\left(E^A_{dr,\tau-1,n'}\right) \Pr(p_\tau, d-i+1) \right] \\
=& \mathbb{1}(n \in \mathcal{N}_{\tau-1,dr}) \Pr\left(E^A_{dr,\tau-1,n} \wedge \sum_{i \in \mathcal{D}^A_{p_\tau}} x_{p_\tau rid} = 0 \right) \\
&+ \sum_{n' \in \{n-1,n\}} \left[\Pr\left(E^A_{dr,\tau-1,n'} \wedge \sum_{i \in \mathcal{D}^A_{p_\tau}} x_{p_\tau rid} = 1 \right) \Pr(p_\tau, d-i+1) \right] \\
=& \mathbb{1}\left(n \in \mathcal{N}_{\tau-1,dr}\right) y^0_{dr,\tau-1,n} + \sum_{i \in \mathcal{D}^A_{p_\tau}} \left\{ \mathbb{1}\left(n-1 \in \mathcal{N}'_{\tau-1,dr}\right) y^1_{dr,\tau-1,n-1,i} \Pr(p_\tau, d-i+1) \right. \\
&\left. + \mathbb{1}\left(n \in \mathcal{N}'_{\tau-1,dr}\right) y^1_{dr,\tau-1,ni} \left[1 - \Pr(p_\tau, d-i+1)\right] \right\}
\end{aligned}$$

The validity of the recent equality originates from the definition of $y^0_{dr,\tau-1,n}$, $y^1_{dr,\tau-1,n-1,i}$ and $y^1_{dr,\tau-1,ni}$ variables. We observe that the left-hand side of constraint (25) is equal to $\Pr(E^A_{dr\tau n})$.

Proof for the variable pairs $q^1_{dr\tau gi}$ ***and*** $q^0_{dr\tau g}$***:*** Similar to the proof for the variable pairs $y^1_{dr\tau ni}$ and $y^0_{dr\tau n}$, we first consider the case $\tau = 0$ as the basis of induction. In constraints (29) and (30), we have $\mathcal{G}_{0,dr} = \{None\}$. Therefore, we can rewrite these two constraints for $\tau = 0$ as follows.

$$q^1_{dr,0,None,i} = \mathbb{1}\left(d \in \mathcal{D}^L_{p_1 i}\right) x_{p_1 rid} + \mathbb{1}\left(d \in \hat{\mathcal{D}}_{p_1 i}\right) x_{p_1 ri,i+L_{p_1}-1} \tag{69}$$
$$\forall d \in \mathcal{D}, r \in \mathcal{R}^D, i \in \mathcal{D}^A_{p_1}$$
$$q^0_{dr,0,None} = 1 - \sum_{i \in \mathcal{D}^A_{p_1}} \left[\mathbb{1}\left(d \in \mathcal{D}^L_{p_1 i}\right) x_{p_1 rid} + \mathbb{1}\left(d \in \hat{\mathcal{D}}_{p_1 i}\right) x_{p_1 ri,i+L_{p_1}-1}\right] \tag{70}$$
$$\forall d \in \mathcal{D}, r \in \mathcal{R}^D$$

Constraints (69) and (70) show that for $\tau = 0$, $q^1_{dr\tau gi}$ and $q^0_{dr\tau g}$ are defined properly. Suppose that variables $q^1_{dr,\tau-1,gi}$ and $q^0_{dr,\tau-1,g}$ are already computed correctly. As the inductive period, we need to show that the model computes the values of variables $q^1_{dr\tau gi}$ and $q^0_{dr\tau g}$ correctly.

Similarly, before discussing constraint (26), we note that the same indexing convention applies to the gender-state variables. The index τ represents the gender state after patients $p_1, \ldots, p_\tau$ have been processed, and the transition from period $\tau - 1$ to period τ is driven by patient p_τ. The variable $q^1_{rd\tau gi}$ is also a joint state-linking variable: it represents the probability that the room gender state in room r on day d is g after patients $p_1, \ldots, p_\tau$ have been processed, and that a bed in the same room is reserved for the next patient $p_{\tau+1}$ admitted on day i. Hence, the index i in $q^1_{rd\tau gi}$ is associated with the admission-day set $\mathcal{D}^A_{p_{\tau+1}}$, whereas patient p_τ appears in the transition terms from period $\tau - 1$ to period τ. With the same indexing convention, the right-hand side of constraint (26) represents the probability that the room gender state in room r on day d by patients p_1 to p_τ is g. Also with respect to constraints (29) and (30), it is clear that only one of the variables $q^0_{dr\tau g}$ and $q^1_{dr\tau gi}, \forall i \in \mathcal{D}^A_{p_{\tau+1}}$ can be nonzero. Also, since constraints (2) and (3) hold, at most one of the variables $q^1_{dr\tau gi}, \forall i \in \mathcal{D}^A_{p_{\tau+1}}$ can be nonzero. Therefore, only one variable that is not equal to 0 appears on the right-hand side of constraint (26).

We use $E^B_{dr\tau g}$ to represent the event that the room gender state in room r on day d by patients p_1 to p_τ is g. To complete the proof, we need to show that $\Pr(E^B_{dr\tau g})$ is equal to the left-hand side of constraint (26). Similarly, when $\sum_{i \in \mathcal{D}^A_{p_\tau}} x_{p_\tau rid} = 1$, the index i in $\Pr(p_\tau, d - i + 1)$ refers to the unique admission day selected for patient p_τ by the fixed

first-stage solution; this case is equivalently decomposed over $i \in \mathcal{D}^A_{p_\tau}$ in the final equality.

$$\begin{aligned}
\Pr(E^B_{dr\tau g}) &= \sum_{\substack{g' \in \mathcal{G}_{\tau-1,dr} | \\ g' \in \mathcal{G}^A_g}} \Pr(E^B_{dr,\tau-1,g'}) \Pr(E^B_{dr\tau g} | E^B_{dr,\tau-1,g'}) \\
&= \sum_{\substack{g' \in \mathcal{G}_{\tau-1,dr} | \\ g' \in \mathcal{G}^A_g}} \Pr(E^B_{dr,\tau-1,g'}) \left[\Pr\left(\sum_{i \in \mathcal{D}^A_{p_\tau}} x_{p_\tau rid} = 1 \right) \Pr\left(E^B_{dr\tau g} | E^B_{dr,\tau-1,g'} \wedge \sum_{i \in \mathcal{D}^A_{p_\tau}} x_{p_\tau rid} = 1 \right) \right. \\
&\quad \left. + \Pr\left(\sum_{i \in \mathcal{D}^A_{p_\tau}} x_{p_\tau rid} = 0 \right) \Pr\left(E^B_{dr\tau g} | E^B_{dr,\tau-1,g'} \wedge \sum_{i \in \mathcal{D}^A_{p_\tau}} x_{p_\tau rid} = 0 \right) \right]
\end{aligned}$$

It is clear that $\Pr\left(E^B_{dr\tau g} | E^B_{dr,\tau-1,g'} \wedge \sum_{i \in \mathcal{D}^A_{p_\tau}} x_{p_\tau rid} = 0\right) = 1$ holds for $g' = g$ if $g \in \mathcal{G}_{\tau-1,dr}$. For other state of g', we have $\Pr\left(E^B_{dr\tau g} | E^B_{dr,\tau-1,g'} \wedge \sum_{i \in \mathcal{D}^A_{p_\tau}} x_{p_\tau rid} = 0\right) = 0$. Also $\Pr\left(E^B_{dr\tau g} | E^B_{dr,\tau-1,g'} \wedge \sum_{i \in \mathcal{D}^A_{p_\tau}} x_{p_\tau rid} = 1\right) = \Pr(p_\tau, d-i+1)$ holds. Considering these points, we can rewrite the above relation as follows.

$$\begin{aligned}
\Pr(E^B_{dr\tau g}) =& \mathbb{1}(g \in \mathcal{G}_{\tau-1,dr}) \Pr\left(E^B_{dr,\tau-1,g}\right) \Pr\left(\sum_{i \in \mathcal{D}^A_{p_\tau}} x_{p_\tau rid} = 0 \right) \\
&+ \sum_{\substack{g' \in \mathcal{G}_{\tau-1,dr} | \\ g' \in \mathcal{G}^A_g}} \left[\Pr\left(\sum_{i \in \mathcal{D}^A_{p_\tau}} x_{p_\tau rid} = 1 \right) \Pr\left(E^B_{dr,\tau-1,g'}\right) \Pr(p_\tau, d-i+1) \right] \\
=& \mathbb{1}(g \in \mathcal{G}_{\tau-1,dr}) \Pr\left(E^B_{dr,\tau-1,g} \wedge \sum_{i \in \mathcal{D}^A_{p_\tau}} x_{p_\tau rid} = 0 \right) \\
&+ \sum_{\substack{g' \in \mathcal{G}_{\tau-1,dr} | \\ g' \in \mathcal{G}^A_g}} \left[\Pr\left(E^B_{dr,\tau-1,g'} \wedge \sum_{i \in \mathcal{D}^A_{p_\tau}} x_{p_\tau rid} = 1 \right) \Pr(p_\tau, d-i+1) \right] \\
=& \mathbb{1}\left(g \in \mathcal{G}_{\tau-1,dr}\right) q^0_{dr,\tau-1,g} + \sum_{i \in \mathcal{D}^A_{p_\tau}} \left\{ \sum_{g' \in \mathcal{G}^A_g | g' \in \mathcal{G}_{\tau-1,dr}} q^1_{dr,\tau-1,g'i} \Pr(p_\tau, d-i+1) \right. \\
&\left. + \mathbb{1}\left(g \in \mathcal{G}_{\tau-1,dr}\right) q^1_{dr,\tau-1,gi} \left[1 - \Pr(p_\tau, d-i+1)\right] \right\}
\end{aligned}$$

The validity of the recent equality originates from the definition of $q^0_{dr,\tau-1,g}$, $q^1_{dr,\tau-1,g'i}$ and $q^1_{dr,\tau-1,gi}$ variables. We observe that the left-hand side of constraint (26) is equal to $\Pr(E^B_{dr\tau g})$.

Therefore, the $SPAS_{SV}$ model is a valid model for the SPAS problem. □

E Hospital data

Table 12 summarizes the characteristics of the LOS data for the selected departments of our collaborating hospital, the People's Hospital of Xinjiang Uygur Autonomous Region in China. The dataset covers the period from September to December 2024, during which 624 patients were admitted. The data include six departments: Abdominal Surgery (ABD), Cardiac Surgery (CAH), Neurosurgery (NCH), Urology (URO), Orthopaedic Surgery (ORT), and Oncology (ONC). For each department, we report the number of specialisms, the number of beds, the number of patients, and the mean and standard deviation of LOS. These departments collectively comprise 36 rooms, 182 beds, and 23 specialties. The LOS data is fitted to a log-normal distribution, which is commonly used (Vancroonenburg et al., 2019). The fixed LOS of each patient is randomly generated based on the fitted log-normal distribution. Overstay lengths are considered for up to three days beyond the fixed LOS, as this range captures the majority of observed overstay events in our dataset. As patient registration time records are unavailable, registration times are generated randomly following a uniform distribution over the planning horizon.

Table 12: Characteristics of hospital data.

Department	Number of specialisms	Number of beds	Number of patients	LOS (days)	
				Mean	Stdev.
Abdominal surg. (ABD)	6	64	213	6.00	10.24
Cardiac surg. (CAH)	5	24	120	16.42	14.83
Neurosurg. (NCH)	3	20	49	13.15	24.55
Urology (URO)	3	16	75	6.81	7.80
Orthopaedic surg. (ORT)	3	34	100	6.33	17.02
Oncology (ONC)	3	24	67	2.80	15.71

F Detailed computational results

F.1 Computational results for RSV, SV and SAA-SV

Table 13 reports computational results obtained by RSV, SV and SAA-SV methods on instances with 1 day and 2 days overstay, respectively. Under the header "Ω", we give the

approximate number of scenarios for each instance, calculated as described in Section 4. Under the header "$Var.$" and "$Con.$", we present the number of variables and constraints in the $SPAS_{SV}$ model. For the RSV method, we record the optimal objective value which also serves as the best lower bound. For the SV and SAA-SV methods, we record the best objective value (Obj), the total computation time to find the best solution, the total computation time when Gurobi either proves the optimality or reaches the time limit, the best lower bound (LB). We compute the percentage gaps $Gap(\%) = (Obj - LB^*)/LB^* \times 100$ of the best objective value found by each method from the best lower bound LB^* among these three methods. The symbol "-" indicates that no feasible solution or no lower bound is found within the time limit.

In terms of lower bounds, the SAA-SV, SV and RSV find the best LBs for 13, 17 and 11 instances, respectively. Specifically, the SAA-SV finds the best LBs primarily in small-scale instances (S-S and S-M with 13 instances), the SV finds the best LBs mainly in medium-scale instances (S-L and M-S with 10 instances), while the RSV finds the best LBs mainly for large-scale instances (M-M with 8 instances). The superior performance of RSV on large-scale instances can be attributed to two main factors. First, RSV simplifies the $SPAS_{SV}$ model into a linear programming formulation, allowing all solution time to be focused on solving it. Second, RSV is set to run for a maximum of 24 hours, which is significantly longer than the 1-hour time limit imposed on the SV and SAA-SV methods. In addition, SV and SAA-SV fail to find lower bounds for 11 and 18 small- and medium-scale instances, respectively. These findings indicate that solving the relaxed $SPAS_{SV}$ model is particularly beneficial in improving the quality of the lower bound.

In terms of best objective values, the SAA-SV method dominates the SV method. Specifically, SAA-SV succeeds in finding feasible solutions for all 64 instances, while the SV method fails for 31 large-scale instances. These two methods find the same best solutions for 8 small-scale instances, of which 3 and 5 instances are proven optimal solutions for SV and SAA-SV, respectively. Moreover, as the size of the instances increases, the obtained gaps range from 0.00% to 261.56% for SV and from 0.00% to 10.20% for SAA-SV. It is worth noting that in Table 13, SAA-SV finds high-quality feasible solutions with an average optimality gap of 3.75% for M-M instances of 500 patients with 1-day overstay. Under this setting ($M = 1$), each patient has $M + 1 = 2$ choices, resulting in a massive scenario space

of $(M+1)^N = 2^{500} \approx 3.27 \times 10^{150}$ scenarios.

Additionally, SAA-SV outperforms SV in terms of computational time. On average, for instances where both methods find feasible solutions with identical objective values, the SAA-SV is 2.67 times faster than SV. For instances where both methods reach the optimal solution, SAA-SV requires only 43% of the time taken by SV to prove optimality. These results suggest that incorporating SAA not only accelerates solution discovery but also improves the efficiency of optimality verification. To conclude, these results demonstrate that SAA-SV is quite effective in terms of both solution quality and computational time.

F.2 Computational results of different scenario sizes

We tested scenario set sizes of 5, 10, 20, and 50 on several representative instance sets (S-S, S-M, S-L, M-S, and M-M) under both 1-day and 2-day overstay settings. We set a 1-hour time limit for each run of the SAA-SV method. Table 15 summarizes the average computational time *Ave. Time* to find the best solution and optimality gap *Ave. Gap* across all instances for each scenario set size in each instance set. From the results, we observe that increasing the scenario set size does not lead to a systematic improvement in solution quality. For small and medium instances (e.g., S-S and S-M), the average optimality gap remains relatively stable when the number of scenarios increases from 5 to 20, with only marginal fluctuations. In contrast, for larger and more challenging instances (e.g., S-L and M-M under 2-day overstay), a larger scenario set may even worsen the solution quality, as reflected by noticeably higher average gaps when $|\Omega^R| = 50$. From a computational perspective, increasing the scenario set size generally results in higher solution times due to the growth of the model size. Overall, these results indicate that using a moderate scenario set size (e.g., $|\Omega^R| = 10$) offers a reasonable trade-off between computational efficiency and solution quality, while excessively large scenario sets provide limited benefits and may negatively impact performance.

F.3 Computational results of *EVP* and *SAA-SV* methods

Tables 16 and 17 present the objective values of the EVP and SAA-SV methods. To compute Obj_{EVP}, we first solve the EVP model and evaluate the resulting solution $(\boldsymbol{x}, \boldsymbol{t})$ in the $SPAS_{SV}$ model, which accounts for an exponential number of scenarios. The value

Table 13: Computational comparisons among *RSV*, *SV* and *SAA-SV* for 1-day overstay instances (best solutions and lower bounds in **bold**, proven optimal in *).

Instance	Ω	*Var.*	*Con.*	*RSV*	*SV*					*SAA-SV*				
					Obj	Time to Best(s)	Time to End(s)	*LB*	*Gap* (%)	*Obj*	Time to Best(s)	Time to End(s)	*LB*	*Gap* (%)
S-S-40		32,984	21,389	6,698.4	**6,861.8***	124.2	467.7	**6,861.8**	0.00	**6,861.8***	1.7	397.4	6,861.8	0.00
S-S-50		49,848	30,288	6,696.5	**6,945.8**	1,214.8	3,600.0	6,889.3	0.00	**6,945.8***	83.8	865.7	**6,945.8**	0.00
S-S-60	1.1×10^{15}	34,552	22,885	6,685.6	**6,775.5***	577.0	680.6	**6,775.5**	0.00	**6,775.5***	292.9	307.3	**6,775.5**	0.00
S-S-70		35,022	24,057	11,170.2	**11,506.0**	3,599.7	3,600.0	11,279.6	0.87	**11,506.0**	488.6	3,600.0	**11,407.3**	0.87
S-M-40		108,329	56,211	11,719.1	12,022.8	1,796.6	3,600.0	**11,859.4**	1.38	**12,015.3**	5.4	3,600.0	11,855.3	1.31
S-M-50		118,719	66,127	17,796.2	**18,031.7**	3,083.0	3,600.0	17,889.8	0.76	**18,031.7**	2,319.5	3,600.0	**17,895.6**	0.76
S-M-60	1.3×10^{30}	96,803	55,695	16,749.0	17,161.8	2,204.8	3,600.0	16,869.2	1.68	**17,148.8**	3,366.5	3,600.0	**16,878.8**	1.60
S-M-70		108,678	65,247	23,304.2	24,658.1	2,353.4	3,600.0	23,502.0	4.92	**24,283.5**	745.2	3,600.0	**23,503.0**	3.32
S-L-40		635,815	251,268	43,356.4	43,692.3	3,548.2	3,600.0	43,395.5	0.63	**43,575.6**	1,276.9	3,600.0	**43,419.3**	0.36
S-L-50		623,616	276,242	45,855.2	49,649.6	3,203.2	3,600.0	**46,004.9**	7.92	**46,900.8**	1,089.3	3,600.0	45,956.8	1.95
S-L-60	1.6×10^{60}	406,517	194,530	66,039.4	70,915.2	3,587.6	3,600.0	**66,422.8**	6.76	**68,231.2**	1,576.4	3,600.0	66,253.4	2.72
S-L-70		570,769	256,261	42,185.0	48,036.5	3,217.1	3,600.0	**42,378.9**	13.35	**43,789.0**	1,792.0	3,600.0	42,317.1	3.33
M-S-40		620,306	386,973	30,730.1	111,536.9	29.3	3,600.0	**30,848.6**	261.56	**31,796.7**	279.6	3,600.0	-	3.07
M-S-50		620,306	386,973	**45,888.1**	114,202.1	25.9	3,600.0	-	148.87	**47,169.2**	778.8	3,600.0	-	2.79
M-S-60	1.8×10^{75}	577,077	356,233	54,678.1	64,973.1	3,432.0	3,600.0	**54,767.4**	18.63	**56,114.0**	1,792.2	3,600.0	-	2.46
M-S-70		755,658	462,863	**46,920.0**	-	-	-	-	-	**48,764.9**	1,450.6	3,600.0	-	3.93
M-M-40		3,144,281	1,486,832	**86,224.6**	235,267.3	148.1	3,600.0	-	172.85	**88,770.5**	1,142.7	3,600.0	-	2.95
M-M-50		2,474,668	1,260,412	**110,328.4**	-	-	-	-	-	**113,936.1**	1,267.5	3,600.0	-	3.27
M-M-60	3.3×10^{150}	2,919,879	1,441,813	**105,272.9**	-	-	-	-	-	**109,727.8**	1,574.5	3,600.0	-	4.23
M-M-70		3,229,928	1,656,556	**95,681.3**	-	-	-	-	-	**100,043.8**	1,808.3	3,600.0	-	4.56
M-L-40		4,706,337	11,794,646	-	-	-	-	-	-	**230,314.4**	3,629.5	7,200.0	-	-
M-L-50		5,265,931	12,366,840	-	-	-	-	-	-	**203,952.3**	3,628.2	7,200.0	-	-
M-L-60	1.1×10^{301}	3,639,401	8,364,660	-	-	-	-	-	-	**221,917.0**	3,613.7	7,200.0	-	-
M-L-70		4,500,194	9,964,022	-	-	-	-	-	-	**242,170.0**	3,616.9	7,200.0	-	-
L-S-40		6,573,459	11,359,356	-	-	-	-	-	-	**121,592.7**	3,729.7	7,200.0	-	-
L-S-50		5,515,547	9,422,927	-	-	-	-	-	-	**152,809.0**	3,790.8	7,200.0	-	-
L-S-60	1.1×10^{301}	6,913,168	11,621,073	-	-	-	-	-	-	**252,757.7**	3,641.9	7,200.0	-	-
L-S-70		4,346,243	7,285,007	-	-	-	-	-	-	**169,263.0**	3,732.0	7,200.0	-	-
L-M-40		20,754,044	43,574,461	-	-	-	-	-	-	**371,267.0**	4,156.8	7,200.0	-	-
L-M-50		22,882,195	47,290,910	-	-	-	-	-	-	**398,662.0**	4,002.7	7,200.0	-	-
L-M-60	1.2×10^{602}	21,935,852	44,504,281	-	-	-	-	-	-	**398,208.0**	3,871.3	7,200.0	-	-
L-M-70		21,101,568	43,388,892	-	-	-	-	-	-	**385,275.0**	3,793.6	7,200.0	-	-

Table 14: Computational comparisons among *RSV*, *SV* and *SAA-SV* for 2-day overstay instances (best solutions and lower bounds in **bold**, proven optimal in *).

Instance	Ω	*Var.*	*Con.*	*RSV*	*SV*					*SAA-SV*				
					Obj	Time to Best(s)	Time to End(s)	*LB*	*Gap* (%)	*Obj*	Time to Best(s)	Time to End(s)	*LB*	*Gap* (%)
S-S-40		38,265	23,770	7,233.8	**7,487.2***	535.7	861.6	**7,487.2**	0.00	**7,487.2***	704.1	1,058.7	**7,487.2**	0.00
S-S-50		56,765	33,612	7,093.5	**7,510.0**	3,600.0	3,600.0	**7,299.1**	2.89	**7,510.0**	7.3	3,600.0	7,296.7	2.89
S-S-60	7.2×10^{23}	40,640	25,866	7,065.5	**7,225.7**	3,044.4	3,600.0	7,205.8	0.00	**7,225.7***	1,998.4	2,812.7	**7,225.7**	0.00
S-S-70		40,367	27,083	11,743.5	12,254.0	3,600.0	3,600.0	11,875.3	2.48	**12,202.8**	2,136.6	3,600.0	**11,957.3**	2.05
S-M-40		127,104	63,189	12,537.8	13,171.0	3,600.0	3,600.0	12,676.0	3.83	**13,155.6**	3,600.0	3,600.0	**12,685.7**	3.70
S-M-50		137,506	74,289	18,907.8	19,300.1	3,600.0	3,600.0	19,001.6	1.46	**19,294.4**	2,484.2	3,600.0	**19,021.7**	1.43
S-M-60	5.2×10^{47}	112,564	62,504	18,090.1	18,926.2	2,114.5	3,600.0	**18,291.4**	3.47	**18,790.4**	1,994.5	3,600.0	18,265.2	2.73
S-M-70		126,577	73,235	24,810.7	27,249.6	1,090.8	3,600.0	**25,012.3**	8.94	**26,439.8**	2,744.5	3,600.0	24,998.1	5.71
S-L-40		739,778	275,602	45,666.7	46,175.2	3,331.1	3,600.0	45,724.9	0.96	**46,062.6**	2,536.7	3,600.0	**45,734.6**	0.72
S-L-50		722,985	306,415	48,349.2	60,961.0	2,572.0	3,600.0	48,455.9	25.81	**50,494.7**	1,088.8	3,600.0	-	4.21
S-L-60	2.7×10^{95}	467,628	215,592	69,860.8	77,023.3	3,513.1	3,600.0	**70,218.5**	9.69	**73,316.3**	1,210.0	3,600.0	70,067.0	4.41
S-L-70		654,485	282,130	45,107.1	56,817.5	2,991.7	3,600.0	**45,248.5**	25.57	**47,660.3**	960.6	3,600.0	-	5.33
M-S-40		834,050	459,281	32,585.9	38,896.6	3,428.1	3,600.0	**32,715.0**	18.90	**34,375.4**	373.9	3,600.0	-	5.08
M-S-50		703,440	427,317	**48,214.3**	119,316.3	3,600.0	3,600.0	-	147.47	**50,052.3**	1,313.8	3,600.0	-	3.81
M-S-60	1.9×10^{119}	659,295	397,177	57,815.4	68,723.3	2,000.4	3,600.0	**57,920.6**	18.65	**60,221.1**	1,825.0	3,600.0	-	3.97
M-S-70		849,857	508,277	49,733.3	64,084.6	3,526.6	3,600.0	**49,811.3**	28.65	**52,368.1**	1,800.1	3,600.0	-	5.13
M-M-40		3,631,575	1,657,379	**91,646.9**	254,317.2	220.2	3,600.0	-	177.50	**95,950.3**	1,676.5	3,600.0	-	4.70
M-M-50		2,839,908	1,403,393	**117,647.8**	-	-	-	-	-	**124,258.5**	1,721.9	3,600.0	-	5.62
M-M-60	3.6×10^{238}	3,347,442	1,603,485	**112,879.0**	-	-	-	-	-	**124,397.1**	394.5	3,600.0	-	10.20
M-M-70		3,698,436	1,848,135	**103,055.9**	-	-	-	-	-	**110,044.9**	1,540.7	3,600.0	-	6.78
M-L-40		5,763,161	14,915,127	-	-	-	-	-	-	**106,557.4**	3,616.5	7,200.0	-	-
M-L-50		6,164,223	15,826,366	-	-	-	-	-	-	**181,046.0**	3,616.8	7,200.0	-	-
M-L-60	1.3×10^{477}	5,247,001	12,594,910	-	-	-	-	-	-	**158,708.0**	3,611.6	7,200.0	-	-
M-L-70		5,175,885	12,626,299	-	-	-	-	-	-	**202,041.1**	3,612.6	7,200.0	-	-
L-S-40		6,350,787	11,538,505	-	-	-	-	-	-	**127,196.9**	3,695.8	7,200.0	-	-
L-S-50		8,899,577	16,114,689	-	-	-	-	-	-	**162,114.3**	3,694.6	7,200.0	-	-
L-S-60	1.3×10^{477}	4,992,009	8,725,029	-	-	-	-	-	-	**175,038.5**	3,652.6	7,200.0	-	-
L-S-70		7,827,045	13,648,802	-	-	-	-	-	-	**205,626.2**	3,611.1	7,200.0	-	-
L-M-40		26,963,530	58,830,269	-	-	-	-	-	-	**334,116.0**	3,795.5	7,200.0	-	-
L-M-50		26,999,921	58,772,182	-	-	-	-	-	-	**290,521.0**	3,771.9	7,200.0	-	-
L-M-60	1.7×10^{954}	20,445,920	43,065,341	-	-	-	-	-	-	**329,312.0**	3,793.9	7,200.0	-	-
L-M-70		18,957,178	39,867,666	-	-	-	-	-	-	**327,627.0**	3,667.5	7,200.0	-	-

Table 15: Impact of scenario set size on computational performance and solution quality of SAA-SV method.

Instance	$\|\Omega^R\| = 5$		$\|\Omega^R\| = 10$		$\|\Omega^R\| = 20$		$\|\Omega^R\| = 50$	
	Ave. Gap(%)	Ave. Time(s)	Ave. Gap(%)	Ave. Time(s)	Ave. Gap(%)	Ave. Time(s)	Ave. Gap(%)	Ave. Time(s)
1-day overstay								
S-S	0.22	540.03	0.22	751.20	0.22	1081.45	0.22	276.31
S-M	1.71	3019.34	1.75	3037.58	1.72	2678.98	1.68	3174.18
S-L	2.14	3358.07	1.94	3092.11	2.06	2650.39	2.02	1900.90
M-S	3.36	1320.68	3.08	1931.32	3.10	2410.85	3.06	1641.36
M-M	3.77	1820.13	4.17	1823.00	4.01	1826.16	4.82	1823.49
2-day overstay								
S-S	1.41	1228.85	1.28	1804.70	1.24	1551.24	1.24	1935.05
S-M	3.36	3099.30	3.35	2536.73	3.42	3110.58	3.45	2671.69
S-L	3.68	2684.90	3.50	3507.75	4.59	3331.39	7.87	2210.49
M-S	4.90	1874.09	4.70	1850.14	4.74	2189.28	6.55	2182.32
M-M	6.41	1835.37	6.43	1865.06	7.28	1843.15	11.30	1830.82

$Obj_{SAA\text{-}SV}$ corresponds to the best objective value obtained by the SAA-SV method. The VSS value is computed as $VSS = (Obj_{EVP} - Obj_{SAA\text{-}SV})/Obj_{SAA\text{-}SV} \times 100$. The symbol "-" indicates that the solution $(\boldsymbol{x}, \boldsymbol{t})$ obtained by solving the EVP model is infeasible when evaluated in the $SPAS_{SV}$ model.

F.4 Contribution of two models in the SAA-SV method

In order to confirm the contribution of the $SPAS_{SAA}$ and $SPAS_{SV}$ models in the SAA-SV method, we further identify the objective values and computational times obtained by each model. Tables 18 and 19 present the gap $Gap(\%)$ and the total computation time $Time$ to find the best solution of each model in each instance. For fair comparison, the Gap values for $SPAS_{SAA}$ are calculated by first applying its solution to the $SPAS_{SV}$ model to obtain the corresponding objective value, which is then used to compute the gap. The $Gap(\%)$ is calculated in the same way as in the previous experiments.

Table 16: Results of *EVP* and *SAA-SV* on 1-day overstay instances with different W^{OP}.

Instance	$W^{OP}=100$			$W^{OP}=1000$			$W^{OP}=2000$		
	Obj_{EVP}	$Obj_{SAA\text{-}SV}$	VSS (%)	Obj_{EVP}	$Obj_{SAA\text{-}SV}$	VSS (%)	Obj_{EVP}	$Obj_{SAA\text{-}SV}$	VSS (%)
S-S-40	6,878.8	6,861.8	0.25	8,683.6	7,117.5	22.00	10,594.4	7,186.1	47.43
S-S-50	7,006.9	6,945.8	0.88	7,922.0	7,101.7	11.55	8,938.8	7,150.5	25.01
S-S-60	6,911.6	6,775.5	2.01	8,423.4	6,870.6	22.60	9,692.8	6,877.4	40.94
S-S-70	11,606.5	11,506.0	0.87	12,693.7	11,907.0	6.61	13,948.1	12,088.5	15.38
S-M-40	12,220.0	12,015.3	1.70	16,954.0	12,538.0	35.22	22,227.9	12,562.1	76.94
S-M-50	18,283.1	18,031.7	1.39	22,973.3	18,231.8	26.01	28,167.0	18,274.7	54.13
S-M-60	17,368.5	17,148.8	1.28	21,823.4	17,459.6	24.99	26,842.4	17,378.7	54.46
S-M-70	24,508.3	24,283.5	0.93	30,069.7	26,343.9	14.14	36,276.3	25,487.6	42.33
S-L-40	43,917.2	43,575.6	0.78	48,795.0	43,629.6	11.84	53,129.5	43,629.7	21.77
S-L-50	47,627.3	46,900.8	1.55	55,335.6	47,491.1	16.52	61,570.0	47,419.3	29.84
S-L-60	69,559.4	68,231.2	1.95	79,019.0	69,081.1	14.39	90,232.7	71,100.5	26.91
S-L-70	44,247.0	43,789.0	1.05	56,930.2	46,951.8	21.25	68,317.4	56,264.1	21.42
M-S-40	32,208.0	31,796.7	1.29	43,437.3	33,451.6	29.85	57,331.8	33,934.0	68.95
M-S-50	48,064.6	47,169.2	1.90	60,791.4	48,411.2	25.57	73,016.6	48,415.1	50.81
M-S-60	56,807.4	56,114.0	1.24	69,263.8	59,132.4	17.13	84,827.3	58,605.2	44.74
M-S-70	50,597.3	48,764.9	3.76	61,260.7	55,459.0	10.46	78,491.0	56,968.5	37.78
M-M-40	91,710.0	88,770.5	3.31	119,850.1	100,320.9	19.47	157,111.1	91,301.3	72.08
M-M-50	116,431.9	113,936.1	2.19	145,139.8	117,336.9	23.69	178,072.1	116,960.0	52.25
M-M-60	115,609.8	109,727.8	5.36	145,133.0	114,370.2	26.90	174,553.2	117,462.6	48.60
M-M-70	101,864.7	100,043.8	1.82	130,695.8	105,326.3	24.09	163,325.2	107,069.1	52.54
M-L-40	109,121.5	106,557.4	2.35	184,214.0	111,689.3	39.37	271,876.3	156,955.8	42.27
M-L-50	187,331.7	181,046.0	3.36	248,077.1	186,329.4	24.89	319,484.8	187,204.1	41.40
M-L-60	162,309.6	158,708.0	2.22	234,604.4	164,973.5	29.68	316,152.8	174,733.9	44.73
M-L-70	212,835.9	202,041.1	5.07	287,999.9	214,679.1	25.46	367,423.6	222,746.9	39.38
L-S-40	131,065.8	127,196.9	2.95	197,176.0	129,264.6	34.44	275,318.7	129,485.8	52.97
L-S-50	168,803.4	162,114.3	3.96	244,059.3	168,551.0	30.94	317,736.3	169,213.8	46.74
L-S-60	182,651.9	175,038.5	4.17	247,685.2	179,874.4	27.38	340,368.3	184,719.2	45.73
L-S-70	214,379.7	205,626.2	4.08	344,867.1	292,311.0	15.24	493,458.3	385,953.7	21.79
L-M-40	387,125.8	334,116.0	13.69	546,949.6	338,864.0	38.04	735,937.7	338,321.0	54.03
L-M-50	352,082.5	290,521.0	17.48	524,315.9	303,707.0	42.08	704,109.5	305,627.0	56.59
L-M-60	379,867.5	329,312.0	13.31	553,528.1	336,584.0	39.19	758,657.3	347,045.0	54.26
L-M-70	387,798.4	327,627.0	15.52	573,584.7	341,965.0	40.38	792,985.9	344,126.0	56.60
Average			3.86			24.73			45.03

Table 18: Model contributions of *SAA-SV* for 1-day overstay instances with different W^{OP}.

Instance	$W^{OP}=100$				$W^{OP}=1000$				$W^{OP}=2000$			
	SAA		SAA-SV		SAA		SAA-SV		SAA		SAA-SV	
	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)
S-S-40	0.00	1.7	0.00	1.7	3.73	0.2	0.00	95.0	18.59	0.2	1.98	628.9
S-S-50	0.14	1.4	0.00	83.8	2.21	8.8	2.21	3,547.0	2.74	7.1	2.60	1,045.0
S-S-60	0.05	3.2	0.00	292.9	7.11	0.3	0.00	1,655.5	4.75	4.2	0.00	493.2
S-S-70	1.16	15.4	0.87	488.6	4.77	22.9	4.18	155.4	8.11	24.3	5.75	969.2
S-M-40	1.31	5.4	1.31	5.4	11.51	6.6	4.79	3,130.2	9.51	10.0	4.96	3,600.0
S-M-50	0.84	0.8	0.76	2,319.5	2.44	5.0	1.65	3,600.0	2.18	3.1	1.88	2,877.0
S-M-60	1.63	24.7	1.60	3,366.5	3.81	25.3	3.13	3,006.4	2.45	36.8	2.45	2,779.0
S-M-70	3.32	745.2	3.32	745.2	20.72	982.6	11.46	2,338.0	7.73	1,358.1	7.73	1,802.0
S-L-40	0.42	18.7	0.36	1,276.9	0.44	10.7	0.44	14.1	0.41	17.5	0.41	20.4
S-L-50	1.95	1,089.3	1.95	1,089.3	3.39	976.5	3.25	3,536.7	2.87	1,726.0	2.87	1,802.6
S-L-60	2.72	1,576.4	2.72	1,576.4	4.43	740.1	4.21	3,078.1	12.76	1,498.4	6.90	3,507.7
S-L-70	3.33	1,792.0	3.33	1,792.0	11.68	1,735.1	10.46	3,419.0	37.69	1,612.6	32.13	3,483.8
M-S-40	3.07	279.6	3.07	279.6	10.17	246.0	8.79	273.2	11.38	188.0	9.93	3,122.3
M-S-50	2.79	778.8	2.79	778.8	5.39	1,466.2	5.32	1,812.0	5.33	498.8	5.33	1,808.0
M-S-60	2.46	1,792.2	2.46	1,792.2	8.14	1,621.4	7.94	1,821.2	7.14	1,800.8	6.78	1,822.1
M-S-70	3.93	1,450.6	3.93	1,450.6	20.37	1,117.1	17.95	1,815.5	21.26	907.0	21.15	1,813.6
M-M-40	2.95	1,142.7	2.95	1,142.7	20.91	1,531.6	16.29	1,860.4	8.34	1,222.9	8.12	1,896.2
M-M-50	3.27	1,267.5	3.27	1,267.5	6.23	1,750.8	6.23	1,840.3	5.92	1,725.2	5.89	1,829.4
M-M-60	4.23	1,574.5	4.23	1,574.5	8.63	1,092.5	8.53	1,828.0	20.38	1,766.7	19.22	1,832.5
M-M-70	4.56	1,808.3	4.56	1,808.3	10.06	1,804.4	9.92	1,828.1	12.15	1,132.6	11.74	1,825.6
Average	2.21	768.4	2.17	1,156.6	8.31	757.2	6.34	2,032.7	10.08	777.0	7.89	1,948.1

Table 17: Results of *EVP* and *SAA-SV* on 2-day overstay instances with different W^{OP}.

Instance	$W^{OP} = 100$			$W^{OP} = 1000$			$W^{OP} = 2000$		
	Obj_{EVP}	$Obj_{SAA\text{-}SV}$	VSS (%)	Obj_{EVP}	$Obj_{SAA\text{-}SV}$	VSS (%)	Obj_{EVP}	$Obj_{SAA\text{-}SV}$	VSS (%)
S-S-40	-	7,487.2	-	11,482.2	9,681.7	18.60	14,883.4	11,549.7	28.86
S-S-50	7,682.5	7,510.0	2.30	12,085.2	8,325.0	45.17	17,019.5	8,349.4	103.84
S-S-60	7,588.7	7,225.7	5.02	10,874.8	7,486.6	45.26	15,098.3	7,647.3	97.43
S-S-70	12,334.9	12,202.8	1.08	15,404.6	14,324.5	7.54	19,594.3	16,069.6	21.93
S-M-40	13,510.6	13,155.6	2.70	24,836.2	14,380.5	72.71	35,470.5	14,362.6	146.96
S-M-50	19,799.8	19,294.4	2.62	28,826.2	20,126.2	43.23	39,555.9	20,082.5	96.97
S-M-60	19,195.2	18,790.4	2.15	28,783.2	20,969.3	37.26	39,369.4	21,602.9	82.24
S-M-70	26,960.9	26,439.8	1.97	40,599.4	35,067.9	15.77	57,562.5	39,931.3	44.15
S-L-40	46,870.5	46,062.6	1.75	57,955.8	46,443.0	24.79	70,307.5	47,268.0	48.74
S-L-50	51,547.0	50,494.7	2.08	71,457.3	53,077.7	34.63	92,738.3	53,770.6	72.47
S-L-60	-	73,316.3	-	-	76,404.7	-	-	77,328.0	-
S-L-70	-	47,660.3	-	75,981.1	57,473.6	32.20	108,857.0	64,541.9	68.66
M-S-40	35,041.9	34,375.4	1.94	59,874.0	36,862.9	62.42	87,883.5	47,200.6	86.19
M-S-50	51,250.1	50,052.3	2.39	73,422.7	53,089.4	38.30	101,105.4	61,490.9	64.42
M-S-60	-	60,221.1	-	-	71,426.2	-	-	79,770.9	-
M-S-70	-	52,368.1	-	-	73,149.7	-	118,642.7	88,347.4	34.29
M-M-40	-	95,950.3	-	-	103,331.6	-	-	103,842.0	-
M-M-50	-	124,258.5	-	-	138,442.3	-	-	147,944.7	-
M-M-60	-	124,397.1	-	-	137,565.7	-	-	139,235.6	-
M-M-70	116,990.7	110,044.9	6.31	182,760.7	135,298.1	35.08	-	147,726.0	-
M-L-40	-	230,314.4	-	-	252,677.1	-	-	275,098.9	-
M-L-50	-	203,952.3	-	-	231,033.6	-	-	254,693.5	-
M-L-60	-	221,917.0	-	-	264,634.4	-	-	291,775.2	-
M-L-70	-	242,170.0	-	-	309,160.9	-	-	355,897.9	-
L-S-40	-	121,592.7	-	-	143,959.4	-	-	166,300.0	-
L-S-50	-	152,809.0	-	-	177,086.0	-	-	198,239.8	-
L-S-60	-	252,757.7	-	-	437,086.1	-	-	641,400.9	-
L-S-70	-	169,263.0	-	-	197,608.7	-	442,247.8	211,276.5	52.23
L-M-40	-	371,267.0	-	-	420,034.0	-	-	469,209.0	-
L-M-50	-	398,662.0	-	-	458,093.0	-	-	511,996.0	-
L-M-60	-	398,208.0	-	-	480,364.0	-	-	536,287.0	-
L-M-70	-	385,275.0	-	-	490,774.0	-	-	693,761.9	-
Average			2.69			36.64			69.96

This table demonstrates that the SAA-SV outperforms the SAA, achieving an average reduction of 2.28% across all instances. Specifically, in 1-day overstay instances, the average gap reduction of SAA-SV over SAA is 0.04% with an additional 388 seconds of computational time for $W^{OP} = 100$, 1.98% with 1275 seconds for $W^{OP} = 1000$, and 2.19% with 1171 seconds for $W^{OP} = 2000$. Similarly, in 2-day overstay instances, the average gap reduction of SAA-SV over SAA is 0.15% with an additional 879 seconds of computational time for $W^{OP} = 100$, 3.38% with 1715 seconds for $W^{OP} = 1000$, and 4.50% with 1625 seconds for $W^{OP} = 2000$. As the instance size increases, the gap reduction achieved by incorporating the $SPAS_{SAA}$ and $SPAS_{SV}$ models decreases. This observation is consistent with the results presented in Section 5.2, due to solving the $SPAS_{SV}$ model becoming increasingly challenging as the instance size grows. The expanding performance difference between SAA-SV and SAA highlights the comparative advantage of the SAA-SV approach over the SAA formulation in these settings. These findings suggest that the SAA-SV method

Table 19: Model contributions of *SAA-SV* for 2-day overstay instances with different W^{OP}.

Instance	$W^{OP}=100$				$W^{OP}=1000$				$W^{OP}=2000$			
	SAA		SAA-SV		SAA		SAA-SV		SAA		SAA-SV	
	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)
S-S-40	0.60	0.2	0.00	704.1	39.14	0.2	16.53	3493.9	58.97	0.2	26.97	3600.0
S-S-50	2.89	7.3	2.89	7.3	20.38	19.1	12.83	3349.7	14.53	44.7	13.13	3173.5
S-S-60	0.30	0.3	0.00	1998.4	7.89	0.3	4.18	2944.4	9.31	0.4	6.26	3526.8
S-S-70	2.29	21.3	2.05	2136.6	16.24	12.7	15.52	1835.0	29.12	27.4	27.06	1566.7
S-M-40	4.24	1.1	3.70	3600.5	21.68	20.4	11.79	3600.0	12.72	35.1	11.34	2583.7
S-M-50	1.60	7.6	1.43	2484.2	10.30	0.7	5.51	3600.0	8.36	26.3	5.28	3236.1
S-M-60	3.56	46.6	2.73	1994.5	25.03	12.5	14.05	3600.0	23.59	60.1	17.53	3600.0
S-M-70	5.81	1299.8	5.71	2744.5	41.28	276.0	39.45	2786.8	-	-	58.34	3179.4
S-L-40	0.82	21.8	0.72	2536.7	1.37	41.1	1.36	3093.1	46.34	7.7	3.15	2535.0
S-L-50	4.21	1088.8	4.21	1088.8	9.41	781.0	9.34	1805.0	10.89	916.1	10.76	1803.9
S-L-60	4.41	1210.0	4.41	1210.0	9.01	1552.4	8.60	3384.2	10.38	1401.1	10.14	1803.4
S-L-70	5.33	960.6	5.33	960.6	25.89	1602.6	25.89	1806.2	41.28	1451.7	41.28	1803.7
M-S-40	5.08	373.9	5.08	373.9	13.13	1488.9	12.99	1837.7	56.70	391.2	44.65	520.6
M-S-50	3.81	1313.8	3.81	1313.8	10.16	1564.6	9.80	1843.1	32.12	798.7	27.16	839.8
M-S-60	3.97	1054.2	3.97	1825.0	22.55	782.8	22.55	1831.9	36.93	690.6	36.87	1817.5
M-S-70	5.13	1800.1	5.13	1800.1	40.72	1801.5	38.51	1844.8	59.11	1661.8	57.90	1828.5
M-M-40	4.70	1676.5	4.70	1676.5	17.96	1802.3	17.87	1927.3	20.06	1800.9	19.81	1915.5
M-M-50	5.62	1721.9	5.62	1721.9	17.95	1803.3	17.74	1872.8	27.42	1251.1	27.04	1865.9
M-M-60	10.20	394.5	10.20	394.5	47.11	594.8	46.65	1883.5	57.67	565.4	56.62	1874.9
M-M-70	6.78	1540.7	6.78	1540.7	32.49	1816.4	30.95	1933.2	47.00	715.7	42.98	1888.6
Average	4.07	727.0	3.92	1605.6	21.49	798.7	18.11	2513.6	31.71	623.5	27.21	2248.2

is particularly well-suited for operational settings where minimizing extra bed usage and ensuring robustness against overstay uncertainty are critical objectives.