The Impact of Estimation: A New Method for Clustering and Trajectory Estimation in Patient Flow Modeling

Chitta Ranjan, Kamran Paynabar

H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology , Atlanta, Georgia, USA,
nk.chitta.ranjan@gatech.edu, kamran.paynabar@isye.gatech.edu

Jonathan E. Helm

Arizona State University, W.P. Carey School of Business, Department of Supply Chain Management , 300 E. Lemon St., Tempe,
AZ 85287, USA, jonathan.helm@asu.edu

Julian Pan

Lean Care Solutions Corporation Pte. Ltd , 28 Ayer Rajah Crescent #03-01, Singapore 139959, Singapore, jp@leancaresolutions.com

The ability to accurately forecast and control inpatient census, and thereby workloads, is a critical and long-standing problem in hospital management. The majority of current literature focuses on optimal scheduling of inpatients, but largely ignores the process of accurate estimation of the trajectory of patients throughout the treatment and recovery process. The result is that current scheduling models are optimizing based on inaccurate input data. We developed a Clustering and Scheduling Integrated (CSI) approach to capture patient flows through a network of hospital services. CSI functions by clustering patients into groups based on similarity of trajectory using a novel semi-Markov model (SMM)-based clustering scheme, as opposed to clustering by patient attributes as in previous literature. Our methodology is validated by simulation and then applied to real patient data from a partner hospital where we demonstrate that it outperforms a suite of well-established clustering methods. Furthermore, we demonstrate that extant optimization methods achieve significantly better results on key hospital performance measures under CSI, compared with traditional estimation approaches, increasing elective admissions by 97% and utilization by 22% compared to 30% and 8% using traditional estimation techniques. From a theoretical standpoint, the SMM-clustering is a novel approach applicable to any temporal-spatial stochastic data that is prevalent in many industries and application areas.

1. Introduction

The mismatch between demand for and supply of medical services caused by high hospital census variability has challenged hospital managers for decades. High census variability is a common problem in hospitals and healthcare centers around the world. This problem leads to poor quality of care, blocking in hospital wards, increase in inpatient length of stay, and ultimately causes significant increase in cost for both patient and hospital (Helm and Van Oyen, 2014). Aiken et al. (2002) studied the effect of overloaded nursing staff induced by census variability and showed its effect on mortality rate, nurse burnout and job dissatisfaction. A common approach to managing census variability in practice involves hospitals procuring excess resources including material, staff, and equipment, leading to frequent instances of underutilization for very expensive resources (Griffin et al. 2012). A better approach is to optimize the utilization of available hospital resources based on patient census estimations. This long-standing problem has been termed the *Hospital Admission Scheduling and Control* (HASC) problem, which can be decomposed into two main steps: *census modeling* (CM) and *resource scheduling* (RS). CM estimates distributional information (typically mean and variance) on patient census at the ward level, which is used as an input to the RS to find the optimal resource allocation plans and schedules for elective inpatient admissions.

A significant body of work addresses the RS through a variety of optimization approaches; however, research on effective census models that integrate with RS is less developed. In this study,

we develop a CM method that integrates well with existing RS methods to solve the HASC. We further demonstrate the importance of the CM component with respect to the outcomes of the RS optimization; a factor that has, to our knowledge, been unaddressed in the current literature. Namely, we show, through computational experiments and a case study based on data from our industry partner, that the CM method typically employed in RS optimization papers leads to markedly inferior optimization results. To conclude this section, we give a short description of the current state of the hospital census forecasting and optimization industry from the experiences of our industry co-author and CEO of a healthcare analytics company. Then we discuss challenges posed by the gaps in CM theory that represent a major hurdle for this burgeoning industry and discuss how our approach seeks to bridge those gaps.

1.1. Real-World Challenges in the Hospital Census Forecasting Industry

Predicting future hospital census levels is a key challenge in the Hospital Admission Scheduling and Control (HASC) problem. Without accurate forecasting mechanisms, controlling the variability in hospital census becomes difficult and creates a major barrier to low-cost, high-quality inpatient care. These consequences of inadequate forecasting are drawn from real-world experience, where our co-author has worked with clients and collaborators globally—Asia, Europe, and North America. All the hospitals he has worked with experience significant mid-week congestion and high levels of blocking.

Current methodologies used in hospitals are ineffective to solving the HASC problem. Almost all the hospitals have lean teams focused on process improvement and some of the bigger hospitals have small analytics teams that use rudimentary models which are ineffective at implementing changes made to solve the HASC problem. All the work done at the hospital level are reactive models (predicting census levels using historical census means, and applying control by canceling surgeries the day before) vs. proactive models (implementing control measures in advance). Recently, some hospitals have been attempting to shift to proactive measures. This has typically involved increasing capacity and lowering utilization, which is cost-prohibitive in the long run. The real solution is to improve the forecasting technology. The methods outlined in this study have proven to be effective on a conceptual level with results shared in the later sections.

Our collaborator, Lean Care Solutions Corporation, is one of the first to provide a patient-level forecasting tool; that is, predicting individual flows

and trajectories of each type of patient entering the hospital. A patient-level forecasting and control tool is imperative for hospitals to effectively solve the HASC problem. While forecasting is the backbone to the solution, Lean Care Solutions also provides the ability for hospitals to create what-if scenarios by modifying admission plans and schedules and to use optimization techniques to customize a dynamic admission plan to minimize blocking and surgical cancelations. This type of analysis and decision support is only possible through patient-level forecasting, as it requires understanding how patient-by-patient modifications to the admission schedule impact hospital census and blocking. This is precisely the type of forecasting that we propose in this study. In fact, workload forecasting is not only useful for bed planning purposes but is also key to allocating resources to the various functions of the hospital. Most notably, workforce planning for front and back end staff accounts for over 50% of hospital costs. Based on the feedback received from Lean Care Solutions' clients, properly allocating staffing reduces various costs, like overtime, and improves staff satisfaction. Overall, it is one key in keeping hospitals profitable and delivering top quality healthcare. After discussing the various needs of the hospitals, it is clear that the key issues in patient flow management, staffing, and scheduling all rely on the critical role of forecasting flexibility and accuracy.

One ongoing challenge for Lean Care Solutions is the issue of defining Patient Types (PTypes) and estimating their probabilistic trajectories over the course of their hospital stay, both of which have a major affect on forecast accuracy. From a computational standpoint, it requires clustering patients into groups, where each group represents one type of patient. Currently, Lean Care Solutions employs various forms of regressions to determine factors to group similar patients together into clusters based on patient characteristics. Many assumptions must be made to fit data into logical PTypes that are scalable and yet give enough information to statistically differentiate patients and enable accurate forecasting. This includes applying numerous heuristics and unfortunately, sacrificing the accuracy of the forecast. At Lean Care Solutions, this process is currently done manually for each hospital, often requiring weeks to months of effort to properly tailor the PTypes for accurate forecasting. These issues of scalability, repeatability, and demonstrated statistical accuracy represent one of the major hurdles for Lean Care Solutions and other participants in the patient-level forecasting space. The methods presented in this study help solve a key problem in parameterizing models for each hospital. Specifically, by clustering patients

based on trajectory (rather than extrinsic characteristics as in current practice), this study significantly improves upon the currently time-consuming and heuristic step of assigning PTypes. Our approach is shown to be scalable, statistically rigorous, accurate, and repeatable. This eliminates the time-consuming, gestalt guess work inherent in current practice and has proven to significantly increase forecast accuracy in addition to improving the results from current decision support methods for admission scheduling.

1.2. Failures of Traditional CM Methods

As noted in Fetter et al. (1980) and Helm and Van Oyen (2014), an appropriate HASC model should have three characteristics: *scalable* to hospitals of any size, consider *ward interactions*, and account for *patient heterogeneity*. Most work in the RS step assumes that patient types are given and uses simple methods for estimating patient trajectories, then employs analytical techniques to capture key hospital metrics in an optimization model. A patient trajectory is characterized by the transitions between wards in a hospital and patient length of stay (LOS) in each ward, and can be expressed as a stochastic function called a location process that maps time to a set of locations—see Figure 1 for an example of two sample path outcomes of a location process.

While the optimization methods are generalizable, the previous approaches to CM for RS optimization lack scalability and are not well suited for capturing patient heterogeneity. These approaches also suffer from the limitation of not properly capturing ward interactions, which is shown to be important in sections 4.2 and 5, where we compare our method with traditional methods that fail to properly account for ward interactions. In this study, we address these issues by developing new methods for clustering

patient location processes based on historical patient flow data.

As an example of how clustering impacts scalability and patient heterogeneity, consider the following. Many traditional approaches to HASC cluster patients by their diagnosis-related group (DRG) or admitting service. However, in working with a large hospital such as our partner hospital, there can be close to 100 such patient types with quite a few of them being very rare. With such a large number of patient types we have found that there is insufficient data to properly estimate patient trajectories even with two or more years of historical data. When further including other important factors such as gender and age, which have been found to be important in determining a patient's trajectory, data scarcity becomes an even larger problem. Current solutions include combining different patient types that are deemed "similar" in order to have sufficient historical data for trajectory estimation. This is a *clustering* problem. Deciding how to combine patient types, however, is a non-trivial effort considering the entire location process (time and location) must be compared to ensure an accurate pairing of two patient types. For example, two patient types may have the same average length of stay (LOS) in the hospital but visit different wards. Another example is if two patients visit similar locations with similar mean LOS, but one has a skewed LOS distribution and the other does not. These factors can all have a significant impact on census forecast accuracy (see Littig and Isken 2007). Because different hospitals have different methods for categorizing patients (different admitting services, DRGs served, etc.), this requires a lengthy and ad-hoc procedure to be performed at each new hospital, significantly impacting scalability. For example, our industry co-author has indicated that this process of clustering under current methods is unique to each hospital and can take months to adequately determine patient types in large hospitals.

A second problem is that once the patient types have been identified, trajectories are assigned based solely on what patient type the patient is identified as. For example, if the patient is a bladder cancer surgery patient their cluster will be bladder cancer surgery. However, other factors that may impact the patient's trajectory and LOS, such as age and gender, cannot be considered after the patient types are defined. This approach is only as good as the granularity of each cluster. However, the clusters are not defined based on the shape of patients' location functions, but rather on other factors available in the data that are believed to be associated with the shape of the location function, but have not been statistically validated. Finally, clusters cannot be too granular or data will be insufficient. This phenomenon impacts both the ability to

Figure 1 Example of Sample Path Outcomes of a Stochastic Location Process for Patient Flow [Color figure can be viewed at wileyonlinelibrary.com]
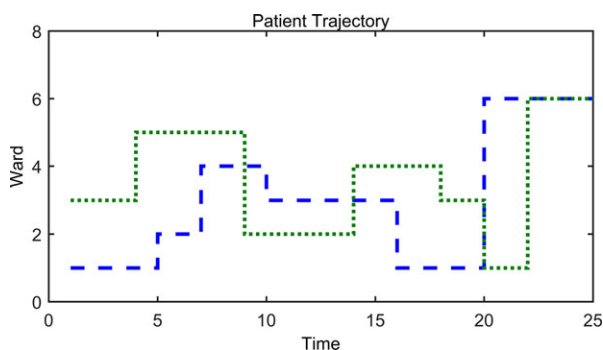
Notes. The *x*-axis shows the time after admission, while the *y*-axis denotes the ward the patient is in at time *t*; each step is a change of ward for the patient.

capture patient heterogeneity and to accurately estimate patient paths because patients are forced into predefined groups rather than assigned a type that most closely matches their projected trajectory.

In contrast, we develop a new clustering approach that clusters patients directly according to similarity of their trajectories (which is what we want to estimate) in a statistically rigorous manner, rather than using these ad-hoc proxies (e.g., DRG, age, and gender). Specifically, we seek to close the gap in the literature by developing new methods for the CM step that provide more effective and scalable clustering of patient types, and a better estimation of the patient trajectories for each patient type. The proposed model, which we call *clustering and scheduling integration* (CSI) is scalable, captures the interactions between hospital wards, and is capable of handling patient heterogeneity. CSI begins with the CM module in which heterogeneous patients are clustered based on the similarity of their trajectories. This provides patient types for accurate estimation of patient trajectories and patient census distributions at the ward level. Finally, these estimates serve as inputs to the RS module to find an optimal hospital resource schedule, which is then shown to outperform the same optimization model using traditional CM methods.

For CM, we propose a novel semi-Markov mixture model (SMM) that integrates the mixture clustering method and semi-Markov models accurately describing stochastic location processes of patient trajectory. To the best of our knowledge, this SMM clustering technique has not been proposed before in the literature, either for the HASC problem or any other problem. The SMM not only clusters patients based on their trajectory but also provides accurate estimates for the trajectory distribution of each group of patients. In the RS module, the output of the CM is fed into an MIP model similar to the model proposed by Helm and Van Oyen (2014) to find the optimal resource schedule for hospitals.

We further show through a case study using real data from a partner hospital that system performance is significantly impacted by the quality of the input from the CM step. In fact, using CSI to parametrize the optimization can enable up to a 50% increase in elective admissions while maintaining the same level of blocking and internal congestion when compared with the same optimization using the traditional estimation approach. Similarly, it is possible to have higher ward utilization compared with traditional CM approaches holding all other metrics constant.

The remainder of the study is organized as follows. We first review the literature and position the study in section 2. Next, we develop the new CSI methodology in section 3, in which the SMM clustering method

for CM is discussed in detail, followed by a brief description of the MIP model used for RS. Then in section 4, we use simulation to validate the proposed CSI model in terms of the accuracy of estimates and the optimality of solutions. In section 5, we apply our CSI methodology in a case study based on historical data from a partner hospital. Finally, in section 6, we conclude the study and discuss future opportunities.

2. Literature

Most existing research in the HASC area has focused on either CM or RS separately. Little work can be found on integrating CM and RS in a cohesive framework. Additionally, existing HASC approaches lack at least one of the aforementioned characteristics of an effective HASC model. The aim of this study was to develop an HASC framework that is scalable, accounts for patient heterogeneity, and considers ward interactions through effective integration of CM and RS.

In the HASC literature, various stochastic and deterministic models have been developed for RS. Green (2006) and Armony et al. (2015) used queuing models to optimize resource scheduling. Ward interactions were not taken into account in either of these papers. Unlike the queuing models, simulation models developed for RS are more flexible and consider the interaction between wards, mostly by using patient pathways between wards in a hospital. Examples of simulation-based models include Hancock and Walter (1979, 1983), Griffith et al. (1976), Jacobson et al. (2006), Harper and Shahani (2002), Zeltyn et al. (2011), and Konrad et al. (2013). However, simulation models are case-specific, cannot be easily generalized or scaled, and rely on the same, less effective PType and path estimation techniques mentioned earlier. Adan et al. (2009), Bekker and Koeleman (2011), and Zhang et al. (2009) used Mixed Integer Programming (MIP) models for optimal RS. These works, however, only focus on either one ward or an isolated feed-forward subset of the hospital, ignoring ward interactions. To address this issue, Helm and Van Oyen (2014) proposed a non-heuristic MIP scheduling model that also used patient pathways to model ward interactions of an entire hospital. Although the RS portion of the model is scalable and considers ward interactions, it does not properly handle patient heterogeneity. Moreover, an empirical method (similar to the traditional method described above) was used to estimate the patient census at the ward level, which we show can degrade the value of the optimal solution.

For RS optimization to be maximally effective, an accurate CM is required to estimate patient arrival rates, their trajectory through the hospital, and, by

combining arrival and trajectory, the patient census at both the ward and hospital levels. Regression analysis and time-series modeling have been widely used for forecasting inpatient admissions and hospital occupancy (Earnest et al. (2005) and Jones et al. (2002)). Abraham et al. (2009) reviewed and compared several models for forecasting daily emergency inpatient admissions and occupancy. They found that the admissions are largely random and hence non-predictable, whereas occupancy can be forecasted using a model combining regression and ARIMA, or a seasonal ARIMA, for up to a week ahead. Their model is capable of forecasting the overall hospital occupancy, but not the occupancy at the ward level. Consequently, it does not account for ward interactions. These approaches are also incapable of capturing what-if scenarios or optimization with respect to inpatient admission decisions. Littig and Isken (2007) used occupancy flow equations to estimate occupancy at different units or wards of a hospital. They predicted patient in- and out-flow using time series and multinomial logistic regression models. They combined these predictions and fed them into a set of flow equations to find the net estimate of the number of patients in a given ward. However, implementing this model in real time presents a major challenge, as even a simple model requires coordination between a variety of real-time data sources and the computational burden of the method is high, so scaling this model to large hospital would be difficult.

To model patient trajectory and LOS, Irvine et al. (1994) and Taylor et al. (2000) proposed a continuous time Markov model for geriatric patients. This model, however, was developed for few wards and lacks scalability. Moreover, the assumption that the LOS at each ward follows the same exponential distribution is not often a good model of reality. Faddy and McClean (1999) used phase-type distributions for patient flow modeling. They interpreted phase-type distributions as a mixture of components (phases) characterized by the severity of patient's illness. Marshall and McClean (2003) extended this idea and developed a model based on Conditional Phase-type distributions combined with a Bayesian network to be able to include a network of inter-related variables representing causality. In phase-type methods, it is assumed that the process begins in the first phase and may either progress through the phases sequentially or enter an absorbing state (see Figure 2a). Consequently, these methods cannot be extended to capture patient trajectories, where patients revisit a ward several times or transition from any ward to any other ward, which is a significant feature according to our data. Thomas (1968) and Kao (1972, 1974) proposed a semi-Markov model to predict recovery progress of coronary patients. This can model any hospital

Figure 2 Illustration of Patient Trajectory Models for a Hospital System
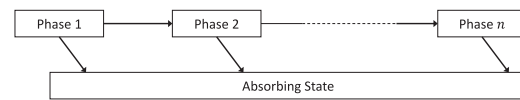
(a) Phase-type model: Patients can transition in a sequential order or leave the system
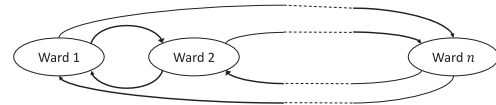
(b) Semi-Markov model: Any back and forth transition from any ward to any ward is possible

system with complicated ward interactions in any direction (See Figure 2b). Thus, this model has *scalability* and can fully model *ward interactions* but is built only for a "homogenous" mix of patients, that is, coronary.

Patient heterogeneity is another challenge in CM. To address this challenge, Helm and Van Oyen (2014) partitioned patients into homogeneous clusters with respect to their diagnosis using DRGs. DRGs have been also used by Fetter et al. (1980) for regional planning. Harper (2005) provided a comprehensive review on clustering techniques, including CART, k-means, neural network, etc. that use more patient attributes (e.g., age, sex, and diagnosis) to find more homogeneous clusters. The main assumption of the DRG and attribute-based methods is that patients who belong to a cluster, follow a similar trajectory. However, this is not necessarily true. Littig and Isken (2007) showed that, patients with similar attributes (e.g., age, sex, diagnosis, etc.) can often have very different trajectories. As an example from our own data, Figure 8a in section 5 shows that although two patients shared the same age, sex, and diagnosis, their trajectories were very different.

In conclusion, the problem of trajectory estimation from a heterogeneous cohort of patients is important. To our knowledge, existing literature fails to address at least one or more challenges among: scalability, ward interaction, and heterogeneity. In the next section, we develop a methodology to address all three challenges and close this gap.

3. Clustering and Scheduling Integrated (CSI) Model for HASC

Figure 3 provides a high-level overview of our methodology. First, historical patient flow data, taken from admit-discharge-transfer (ADT) records, are used to group the patients based on their trajectory using a semi-Markov Mixture (SMM) model-

based clustering approach. The parameters for the semi-Markov processes of patient trajectory for each cluster are estimated as a part of the clustering process. These stochastic location processes are then combined with a model of the non-stationary patient arrival process to form a stochastic process (a Poisson arrival-location model or PALM, see Massey and Whitt (1993)) that captures the ward-network census. Estimation of this stochastic network census process enables the derivation of three important products for hospital managers: (1) Descriptive: accurate census forecasting, (2) What-if scenarios: impact of potential modifications to admission schedules, and (3) Prescriptive: MIP-based admission scheduling optimization.

3.1. Semi-Markov Mixture (SMM) Clustering for Modeling Patient Trajectories

When a new patient arrives to the hospital, they are initially assigned a bed in a hospital ward. The patient stays at that ward for a stochastic duration and then transfers to another ward or is discharged from the hospital. This process repeats if the patient is transferred to another ward of the hospital.

A general hospital serves a cohort of many different types of patients. Each type of patient requires different services during their hospital stay. The first task is to identify patient types through clustering. As mentioned previously, conventional clustering methods are not applicable to this problem due to the fact two patients with the same observed attributes often have different trajectories.

To manage the heterogeneous mix of patients in a hospital, we develop a semi-Markov mixture model for clustering based on patient trajectory rather than predefined groupings based on patient attributes. Patients in each cluster are assumed to follow a semi-Markovian trajectory through the hospital, which has been validated in the literature (e.g., Hancock and Walter (1983)). The SMM produces three important products that significantly improve the generality and scalability of our method: (1) appropriate patient

groupings based on trajectory, (2) the optimal number of patient types, and (3) accurate trajectories for each patient type. In section 5, we show that this approach yields more efficient patient clusters and more accurate trajectory models than traditional approaches. Moreover, to the best of our knowledge, there is no existing approach for developing a semi-Markov mixture model and using it for clustering spatial-temporal data.

3.1.1. SMM Model Structure. Let $\mathcal{K}$ be the set of unknown patient types, where each patient type's trajectory follows a unique semi-Markov process. The population of patient trajectory data, thus, follows a mixture of an unknown number, $|\mathcal{K}|$, of semi-Markov processes. Each mixture component, which we call a *cluster* henceforth, has a different semi-Markov process distribution. The first step is to determine the set of clusters ($\simeq \mathcal{K}$) and estimate their corresponding trajectory distributions.

Consider a sample of trajectory data for N patients observed over a maximum time period of length T. Time is measured by discrete units, for example, a day, quarter of day, hour, to be chosen depending on the desired granularity. The set of possible lengths of stay is denoted by $\mathcal{T} = \{1, 2, \ldots, T\}$. Let $\mathcal{U} = \{\underline{\mathcal{U}}, \bar{\mathcal{U}}\}$ denote the set of all states (wards) where $\underline{\mathcal{U}}$ is the set of all transient states and $\bar{\mathcal{U}}$ is the set of all absorbing states. The first state when the patient enters the system (hospital) is called the initial state and the last state, which is an absorbing state, indicates a patient's end of stay in the form of discharge or death. All the states during the patient's hospital stay are transient states. The set of initial and transient states are the same, as a patient may enter the hospital at any arbitrary location.

A patient n's ($n \in N$) trajectory is represented as $\mathbf{y}^{(n)} = (\{u_1, v_1\}, \ldots, \{u_{L^{(n)}}, v_{L^{(n)}}\}, \{\bar{u}\})$, where $u_l \in \underline{\mathcal{U}}$ indicates the visited ward, $v_l \in \mathcal{T}$ is the length of stay at the corresponding ward, $\bar{u} \in \bar{\mathcal{U}}$ is the absorbing state from where the patient leaves the hospital, and subscript l, $l = 1, 2, .., L^{(n)}$, indicates the sequence of ward visits (*state* and *ward* are used synonymously in

Figure 3   Clustering and Scheduling Integrated (CSI) Model Overview
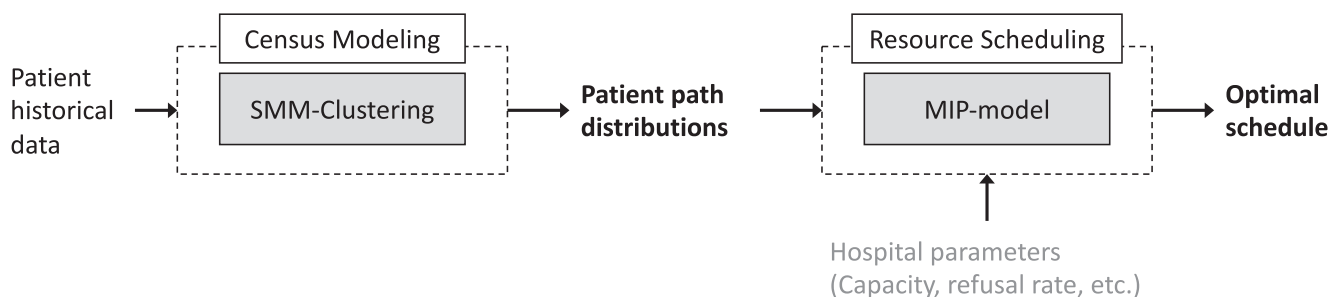
this study). $L^{(n)}$ is the patient n's path sequence length. This model can capture general network behavior, as there is no restriction on the number of times a patient can visit any particular ward.

We formulate the problem by defining a set of parameters, $\boldsymbol{\Theta} = \{\Theta^{(k)}\}$, $k \in \mathcal{K}$. Each $\Theta^{(k)}$ is comprised of the mixture weight, $\pi^{(k)}$, and semi-Markov process parameters, $\{\rho^{(k)}, P^{(k)}, H^{(k)}\}$, for the k-th mixture. The mixture weight, $\pi^{(k)}$, denotes the probability of a randomly chosen patient belonging to cluster k. Letting $z^{(n)}$ be a hidden variable representing the cluster index for patient n, then the mixture weight can be expressed as, $\pi^{(k)} = p_{\boldsymbol{\Theta}}(z = k)$. Also, $\Sigma_{k \in \mathcal{K}} \pi^{(k)} = 1$.

Of the remaining mixture parameters, $\boldsymbol{\rho}^{(k)} = \{\rho_u^{(k)}\}$, $u \in \mathcal{U}$, denotes the initial state probability. It can be expressed as $\rho_u^{(k)} = p_{\boldsymbol{\Theta}}(u_1 = u | z = k)$, the probability of the first state of a patient trajectory being ward u given the patient belongs to cluster k. The matrix $\mathbf{P}^{(k)} = [P_{uj}]$, $u, j \in \mathcal{U}$ is the transition probability matrix, where $P_{uj}^{(k)} = p_{\boldsymbol{\Theta}}(u_l = j | u_{l-1} = u, z = k)$, the probability of transitioning from ward u to j for a patient in cluster k. Finally, $\mathbf{H}^{(k)} = [H_{uj}^{(k)}(v)]$, $u, j \in \mathcal{U}$, $v \in \mathcal{T}$ is a three-dimensional tensor representing the holding mass distribution, where $H_{uj}^{(k)}(v) = p_{\boldsymbol{\Theta}}(v_l = v | u_l = j, u_{l-1} = u, z = k)$ gives the probability of a patient in cluster k spending v time units in ward u before transitioning to ward j (from u). As $\{\rho^{(k)}, P^{(k)}, H^{(k)}\}$ are probability distributions, the following hold:

$$\sum_{u \in \mathcal{U}} \rho_u^{(k)} = 1, \sum_{j \in \mathcal{U}} P_{uj}^{(k)} = 1, and \sum_{v \in \mathcal{T}} H_{uj}^{(k)}(v) = 1. \quad (1)$$

Using this parameterization, we represent the conditional probability of any patient n's trajectory, $\mathbf{y}^{(n)}$, given it is generated by cluster k, in Equation (2). The first part of the equation is the initial state probability. The terms inside the product is the transition probability times the holding time probability corresponding to the transition the patient made, and the amount of time the patient spent at the ward before transitioning.

$$p_{\boldsymbol{\Theta}}(\mathbf{y}^{(n)} | z^{(n)} = k) = p(u_1 | \rho^{(k)}) \prod_{l=1}^{L^{(n)}} p(u_{l+1} | u_l; \mathbf{P}^{(k)})$$
$$p(v_l | u_{l+1}, u_l; \mathbf{H}^{(k)})$$
$$= \rho_{u_1}^{(k)} \prod_{l=1}^{L^{(n)}} \left\{ P_{u_l, u_{l+1}}^{(k)} \cdot H_{u_l, u_{l+1}}^{(k)}(v_l^{(i)}) \right\}. \quad (2)$$

Consequently, by considering the probability of belonging to each cluster, k, the probability distribution function (pdf) of the SMM model with $\mathcal{K}$ components is written as

$$p(\mathbf{y}^{(n)} | \boldsymbol{\Theta}) = \sum_{k \in \mathcal{K}} p_{\boldsymbol{\Theta}}(z^{(n)} = k) p_{\boldsymbol{\Theta}}(\mathbf{y}^{(n)} | z^{(n)} = k)$$
$$= \sum_{k \in \mathcal{K}} \pi^{(k)} \left[\rho_{u_1}^{(k)} \prod_{l=1}^{L^{(n)}} \left\{ P_{u_l, u_{l+1}}^{(k)} \cdot H_{u_l, u_{l+1}}^{(k)}(v_l) \right\} \right]. \quad (3)$$

Given an i.i.d. sample of N patient trajectories, $\mathbf{Y} = \{\mathbf{y}^{(n)}; n = 1, \ldots, N\}$, the likelihood function is, thus, given by

$$p_{\boldsymbol{\Theta}}(\mathbf{Y}) = \prod_{n=1}^{N} p(\mathbf{y}^{(n)} | \boldsymbol{\Theta})$$
$$= \prod_{n=1}^{N} \sum_{k \in \mathcal{K}} \pi^{(k)} \left[\rho_{u_1}^{(k)} \prod_{l=1}^{L^{(n)}} \left\{ P_{u_l, u_{l+1}}^{(k)} \cdot H_{u_l, u_{l+1}}^{(k)}(v_l) \right\} \right]. \quad (4)$$

The parameters of the SMM mixture model, $\boldsymbol{\Theta}$, can be estimated by maximizing the (log)likelihood function in Equation (4). However, if there is no observed transition between any two states or no instance of any particular length of stay, the likelihood function becomes zero. To avoid this issue, we use a Bayesian approach that assigns very small prior probabilities to all model parameters, denoted by $p(\boldsymbol{\Theta})$. Thus, according to Bayes rule, the posterior probability for $\boldsymbol{\Theta}$ can be expressed as $p(\boldsymbol{\Theta} | \mathbf{Y}) = \frac{p(\mathbf{Y} | \boldsymbol{\Theta}) p(\boldsymbol{\Theta})}{p(\mathbf{Y})}$. Since $p(\mathbf{Y})$ is independent of $\boldsymbol{\Theta}$, it suffices to maximize the non-normalized posterior log-likelihood in Equation (5) to obtain the optimal $\boldsymbol{\Theta}^*$, also known as the *maximum a posteriori* (MAP) estimates of $\boldsymbol{\Theta}$.

$$\boldsymbol{\Theta}^* = \arg \max_{\boldsymbol{\Theta}} \log\{p(\mathbf{Y} | \boldsymbol{\Theta}) p(\boldsymbol{\Theta})\}. \quad (5)$$

The optimization problem in Equation (5) does not have a closed-form solution. Furthermore, the non-normalized posterior log-likelihood function is non-convex so Equation (5) cannot be solved using standard convex optimization methods. As a result, we develop an iterative *expectation-maximization* (EM) procedure in the following section to obtain the parameter estimates.

3.1.2. Parameter Estimation via Expectation-Maximization (EM). An Expectation-Maximization (EM) algorithm is an effective approach for learning maximum likelihood or maximum a posteriori (MAP) estimates, where the likelihood is a function of unobserved latent variables (in our case, z). It is an iterative approach comprising of an Expectation (E-step) and Maximization (M-step) in each iteration. In the E-step of any iteration p, we obtain a lower bound on the objective function by taking its expectation at the

current parameter estimate, $\boldsymbol{\Theta}^{(p)}$. Then, in the M-step, we re-estimate the parameters (update), to obtain $\boldsymbol{\Theta}^{(p+1)}$, that maximizes the expectation from E-step. This procedure results in an increase of the likelihood function with guaranteed convergence under some weak regularity conditions that are satisfied in most practical situations (Wu 1983). The specific EM algorithm we develop for the SMM mixture model is as follows: **E-step**

We find the expected value of the maximum a posteriori function in Equation (5) with respect to the current parameter estimate, $\boldsymbol{\Theta}^{(p)}$, denoted by $Q(\boldsymbol{\Theta}\mid\boldsymbol{\Theta}^{(p)})$ in Equation (6).

$$Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(p)}) = \mathbb{E}_{\boldsymbol{\Theta}^{(p)}}[\log(p(\mathbf{Y}|\boldsymbol{\Theta})p(\boldsymbol{\Theta})]. \qquad (6)$$

For a simpler expression of the Q function in Equation (6), we define a *membership* probability distribution. Membership probability, denoted by Ω_{nk}, is the probability of observing any patient n's trajectory, $\mathbf{y}^{(n)}$, generated by cluster k, given parameters $\boldsymbol{\Theta}$ (see Equation (8)).

$$\Omega_{nk}(\boldsymbol{\Theta}) = \frac{\pi^{(k)}p_{\boldsymbol{\Theta}}(\mathbf{y}^{(n)}|z^{(n)}=k)}{\sum_{k'\in\mathcal{K}}\pi^{(k')}p_{\boldsymbol{\Theta}}(\mathbf{y}^{(n)}|z^{(n)}=k')}, \qquad (7)$$

$$\Omega(\boldsymbol{\Theta}) = [\Omega_{nk}(\boldsymbol{\Theta})]; \ n = 1,\ldots,N, k \in \mathcal{K}. \qquad (8)$$

The Q function, can thus be expressed as,

$$\begin{aligned} Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(p)}) =& \mathbb{E}_{\boldsymbol{\Theta}^{(p)}}\left[\log(p(\mathbf{Y}|\boldsymbol{\Theta})p(\boldsymbol{\Theta})\right] \\ =& \sum_{n=1}^{N}\sum_{k\in\mathcal{K}}\Omega_{nk}(\boldsymbol{\Theta}^{(p)})\log\left[\pi^{(k)}p_{\boldsymbol{\Theta}}(\mathbf{y}^{(n)}|z^{(n)}=k)\right]. \\ &+ \log p(\boldsymbol{\Theta}) \end{aligned}$$
$$(9)$$

M-step

In the *maximization* step, the parameters that maximize the Q function are estimated. The updated parameters are, thus,

$$\boldsymbol{\Theta}^{(p+1)} = \arg\max_{\boldsymbol{\Theta}}\left\{Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(p)})\right\}. \qquad (10)$$

To solve Equation (10), we will estimate the posterior of the parameters using a Dirichlet prior probability distribution for $\boldsymbol{\Theta}$, $p(\boldsymbol{\Theta})$. The Dirichlet distribution is chosen because (1) the parameters of a first-order semi-Markov mixture are in the form of multinomial probabilities, which are suitably represented by Dirichlet distribution, and (2) the conjugate of Dirichlet is also a Dirichlet distribution, thus posterior computation is straightforward.

For any set of multinomial parameters, $x = (x_1, \ldots, x_m)$, such that $\Sigma_{i=1}^{m}x_i = 1, 0 \le x_i \le 1$, a Dirichlet distribution is given by,

$$p(x_1,\ldots,x_m|a_1,\ldots,a_m) = \frac{1}{B(a)}\prod_{i=1}^{m}x_i^{a_i-1}, \qquad (11)$$

where a_i's are hyperparameters for x, and $B(a) = \frac{\prod_{i=1}^{m}\Gamma(a_i)}{\Gamma(\Sigma_{i=1}^{m}a_i)}$, a constant factor for the Dirichlet probability distribution function. Using the prior probability distributions, assumption of independence of parameters, and plugging Equation (2) into Equation (9), we obtain the posterior distributions. We show in Appendix A, the posterior distributions are Dirichlet, and how to update parameters to maximize Equation (9). The derived expressions are shown below,

$$\pi^{(k)} = \frac{\sum_{n=1}^{N}\Omega_{nk}(\boldsymbol{\Theta}^{(p)}) + a_{\pi}^{(k)}}{\sum_{k'\in\mathcal{K}}\left[\sum_{n=1}^{N}\Omega_{nk'}(\boldsymbol{\Theta}^{(p)}) + a_{\pi}^{(k')}\right]}, \forall k \in \mathcal{K}.$$

$$\rho_u^{(k)} = \frac{\sum_{n=1}^{N}\Omega_{nk}(\boldsymbol{\Theta}^{(p)})\kappa(u_1,u) + a_{\rho,u}^{(k)}}{\sum_{u'\in\mathcal{U}}\left[\sum_{n=1}^{N}\Omega_{nk}(\boldsymbol{\Theta}^{(p)})\kappa(u_1,u') + a_{\rho,u'}^{(k)}\right]},$$
$$\forall u \in \mathcal{U}, k \in \mathcal{K},$$

$$P_{uj}^{(k)} = \frac{\sum_{n=1}^{N}\Omega_{nk}(\boldsymbol{\Theta}^{(p)})\bar{\kappa}_{uj}(\mathbf{y}^{(n)}) + a_{P,uj}^{(k)}}{\sum_{j'\in\mathcal{U}}\left[\sum_{n=1}^{N}\Omega_{nk}(\boldsymbol{\Theta}^{(p)})\bar{\kappa}_{uj'}(\mathbf{y}^{(n)}) + a_{P,uj'}^{(k)}\right]},$$
$$\forall u,j \in \mathcal{U}, k \in \mathcal{K},$$

$$H_{uj}^{(k)}(v) = \frac{\sum_{n=1}^{N}\Omega_{nk}(\boldsymbol{\Theta}^{(p)})\tilde{\kappa}_{uj,v}(\mathbf{y}^{(n)}) + a_{H,uj}^{(k)}(v)}{\sum_{v'\in\mathcal{T}}\left[\sum_{n=1}^{N}\Omega_{nk}(\boldsymbol{\Theta}^{(p)})\tilde{\kappa}_{uj,v'}(\mathbf{y}^{(n)}) + a_{H,uj}^{(k)}(v')\right]},$$
$$\forall u,j \in \mathcal{U}, v \in \mathcal{T}, k \in \mathcal{K}.$$

3.1.3. SMM-Clustering Algorithm. In this section, we detail the specific algorithm for implementing SMM clustering (see Algorithm 1) and discuss key features such as sensitivity to initialization, identifiability, computational complexity, and optimization acceleration techniques. Additionally, an alternative algorithm, which is approximate but faster, is given in Appendix B.

As shown in Algorithm 1, we take the trajectory data and the number of clusters as inputs. The estimation procedure is initialized by randomly initializing Ω such that the memberships of a patient among all clusters is equal to 1. The Dirichlet prior hyperparameters, $a_{(\cdot)}$, are chosen as a small number and uniform for all parameters (ε is a small positive number, taken as $1e-5$ in our experiments). Thereafter, iterative estimation is done, where the membership probabilities and the SMM parameters are updated in each iteration.

In our implementation, we set a termination condition so that the algorithm terminates after maxIter

iterations. Iterations can also be performed until a given measure of convergence. For example, convergence can be measured in terms of either no change in the objective function or hard cluster assignments of the trajectories—that is, the clusters do not change much between iterations. Tracking of cluster reassignments (in each iteration) works better than tracking the objective function, as the change in latter becomes extremely small after few iterations. But, in practice, having an upper bound for the number of iterations (maxIter) is more useful due to potential identifiability issues. Especially when the data size is large, it can take a very long time for cluster reassignments of all trajectories to stabilize between iterations. maxIter serves as a reasonable trade-off between computation time and accurate results, and hence is commonly employed in many clustering implementations.

The runtime computational complexity of the algorithm is linear in the length of sequences, the sample size, the number of clusters, and the number of iterations, that is, $O(\text{maxIter} \times KNL)$, where L is average sequence length. This linear complexity makes the implementation fast. Additionally, several steps in the algorithm can be vectorized, for example, parameter normalization, for increased speed. Computation time can be further reduced by: (1) *parallelization:* since the parameter update equations are independent for each cluster, state, and length-of-stay, we can split the computation across many computing nodes; and (2) *stochastic clustering:* using a random subsample of data for parameter updating in each iteration. Parallelization requires multiple computing nodes, while the stochastic clustering is suitable when the data sample is very large. A higher order Markov clustering extension of our proposed model will increase the computational complexity, and therefore may require some or all of the above techniques for tractable solution times.

Algorithm 1. SMM-Clustering Algorithm

Input: Trajectory data, $\mathbf{Y}=\{\mathbf{y}^{(n)}; n=1,\ldots,N\}$, number of clusters, K.
  Initialize:
    $\Omega \leftarrow$ random matrix$_{N \times K}$ ▷ s.t. sum of each row is equal to 1
    $a_\pi \leftarrow \epsilon/K; a_\rho \leftarrow \frac{\epsilon}{|\mathcal{U}| \times K};$
    $a_\mathbf{P} \leftarrow \frac{\epsilon}{|\mathcal{U}| \times |\mathcal{U}| \times K}; a_\mathbf{H} \leftarrow \frac{\epsilon}{|\mathcal{U}| \times |\mathcal{U}| \times |\mathcal{U}| \times K}$ ▷ Prior hyperparameters
  for iter =1,…, maxIter **do**
    $\Theta \leftarrow$ SMMParameters(Ω)
    $\Omega \leftarrow$ MembershipProb(Θ)
  end for
  function MembershipProb(Θ)
    $\Omega \leftarrow \mathbf{0}_{N \times K}$
    for k=1,…,K **do**
      for n=1,…,N **do**
        Fetch trajectory sequence, $\mathbf{y}^{(n)} = (\{u_1, v_1\}, \ldots, \{u_{L^{(n)}}, v_{L^{(n)}}\}, \{\bar{u}\})$
        $\Omega_{n,k} \leftarrow \pi^{(k)} * \rho_{u_1}^{(k)}$
        for $l = 1, \ldots, L^{(n)}$ **do**

(continued)

$$\Omega_{n,k} \leftarrow \Omega_{n,k} * \mathbf{P}_{u_l, u_{l+1}}^{(k)} * \mathbf{H}_{u_l, u_{l+1}}^{(k)}(v_l)$$
        end for
      end for
    end for
    $\Omega_{n,k} \leftarrow \frac{\Omega_{n,k}}{\sum_{k'=1}^{K} \Omega_{n,k'}}; k = 1, \ldots, K, n = 1, \ldots, N$ ▷ Normalizing for, $\sum_{k=1}^{K} \Omega_{n,k} = 1, \forall n = 1, \ldots, N$
  return Ω
  end function
  function SMMParameters (Ω)
    $\pi^{(k)} \leftarrow a_\pi + \sum_{n=1}^{N} \Omega_{n,k}; k = 1, \ldots, K$
    $\pi^{(k)} \leftarrow \pi^{(k)} / \sum_{k'=1}^{K} \pi_{k'}$ ▷ Normalizing
    $\rho \leftarrow \mathbf{0}_{\mathcal{U} \times K}; \mathbf{P} \leftarrow \mathbf{0}_{\mathcal{U} \times \mathcal{U} \times K}; \mathbf{H} \leftarrow \mathbf{0}_{\mathcal{U} \times \mathcal{U} \times \mathcal{T} \times K}$
    $\rho_u^{(k)} \leftarrow a_\rho; u \in \mathcal{U}, k = 1, \ldots, K$
    $\mathbf{P}_{u,u'}^{(k)} \leftarrow a_\mathbf{P}; u, u' \in \mathcal{U}, u \neq u', k = 1, \ldots, K$
    $\mathbf{H}_{u,u'}^{(k)}(v) \leftarrow a_\mathbf{H}; u, u' \in \mathcal{U}, u \neq u', v \in \mathcal{T}, k = 1, \ldots, K$
    for k=1,…,K **do**
      for n=1,…,N **do**
        Fetch trajectory sequence, $\mathbf{y}^{(n)} = (\{u_1, v_1\}, \ldots, \{u_{L^{(n)}}, v_{L^{(n)}}\}, \{\bar{u}\})$
        $\rho_{u_1}^{z^{(n)}} \leftarrow \rho_{u_1}^{z^{(n)}} + \Omega_{n,k}$
        for $l = 1, \ldots, L^{(n)}$ **do**
          $\mathbf{P}_{u_l, u_{l+1}}^{z^{(n)}} \leftarrow \mathbf{P}_{u_l, u_{l+1}}^{z^{(n)}} + \Omega_{n,k}$
          $\mathbf{H}_{u_l, u_{l+1}}^{z^{(n)}}(v_l) \leftarrow \mathbf{H}_{u_l, u_{l+1}}^{z^{(n)}}(v_l) + \Omega_{n,k}$
        end for
      end for
    end for ▷ Normalizing as per Eq. 1
    for k=1,…,K **do**
      $\rho_u^{(k)} \leftarrow \rho_u^{(k)} / \sum_{u' \in \mathcal{U}} \rho_{u'}^{(k)}; \forall u \in \mathcal{U}, k = 1, \ldots, K$
      $\mathbf{P}_{u,j}^{(k)} \leftarrow \mathbf{P}_{u,j}^{(k)} / \sum_{j' \in \mathcal{U}} \mathbf{P}_{u,j}^{(k)}; \forall u, j \in \mathcal{U}, k = 1, \ldots, K$
      $\mathbf{H}_{u,j}^{(k)}(v) \leftarrow \mathbf{H}_{u,j}^{(k)}(v) / \sum_{v' \in \mathcal{T}} \mathbf{H}_{u,j}^{(k)}(v'); \forall u, j \in \mathcal{U}, v \in \mathcal{T}, k = 1, \ldots, K$
    end for
    return $\Theta = \{\pi^{(k)}, \rho^{(k)}, \mathbf{P}^{(k)}, \mathbf{H}^{(k)}; k=1, \ldots, K\}$
  end function
Output: $\hat{\Theta} = \{\hat{\Theta}^{(k)}\}, k = 1, \ldots, K$; and cluster soft assignments Ω.

Similar to most other clustering methods, the SMM-clustering results are sensitive to the initialization. While some experts suggest using prior system knowledge for initialization, such as DRGs, we feel such an initialization may introduce bias into our results; particularly because a main motivation for developing this approach was that DRG clustering was not sufficiently accurate in practice. We, therefore, recommend random cluster initialization. To avoid potentially poor solutions resulting from a particular initialization, we perform multiple runs with different random initializations and choose the solution with the highest final objective function.

3.1.4. Determining the Number of Clusters. To determine the appropriate number of clusters, we estimate the SMM model and compute the Q function, which is analogous to the likelihood. We then increase the number of clusters, $|\mathcal{K}|$, by one at each iteration. We stop when there is no significant change in the Q function by adding an additional cluster (popularly known as the *elbow* method). To eliminate redundant clusters, we perform pairwise hypothesis tests with controlled type-I error for the identified clusters. We use the Chi-square hypothesis test developed by Billingsley (1960, 1961) for comparing transition

probabilities and Kolmogorov–Smirnov for comparing the distributions on the initial state and the holding time. We merge any clusters that are found similar by these tests and then perform the tests again in iterative fashion until no redundant clusters are detected. A similar approach for removing redundant clusters was used by Weiss et al. (1982).

3.1.5. Trajectory Estimation for Each Cluster. After parameter estimation, the next step is to estimate the patient trajectory distributions which are characterized by the visited wards and length of stay at each ward. Using the selected number of clusters and corresponding semi-Markov process estimates from our EM algorithm, we compute the probability distribution of patient trajectory, denoted by $\Gamma(d) = [\gamma_j^{(k)}(d)]$; $j \in \mathcal{U}$, $k \in \mathcal{K}$ and $d = 1, 2, \ldots$, where $\gamma_j^{(k)}(d)$ is the probability that a patient of cluster k is in ward j after d days (we use a day as a time unit for v). This distribution is one of the key inputs to the scheduling optimization.

To estimate $\Gamma(d)$ we use *interval transition probabilities*, $\Phi^{(k)} = [\phi_{uj}^{(k)}(d)]$; $u, j \in \mathcal{U}$, $k \in \mathcal{K}$ and $d = 1, 2, \ldots$, where $\phi_{uj}^{(k)}(d)$ is the probability that a patient in cluster k is in ward j on day d, given that the patient entered the hospital in ward u. Recalling that, for a type k patient, $H_{uj}^{(k)}(d)$ is the holding time probability distribution in ward u before transitioning to ward j and $P_{uj}^{(k)}$ is the probability of transitioning from ward u to j, then $\phi_{uj}^{(k)}(d)$ is computed as

$$\phi_{uj}^{(k)}(d) = P_{uj}^{(k)}H_{uj}^{(k)}(d) + \delta_{uj}\sum_{l\in\mathcal{U}\setminus\{u\}}\sum_{d'=d+1}^{\infty}P_{ul}^{(k)}H_{ul}^{(k)}(d')$$
$$+ \sum_{l\in\underline{\mathcal{U}}\setminus\{j\}}\sum_{d'=1}^{d}P_{ul}^{(k)}H_{ul}^{(k)}(d')\phi_{lj}^{(k)}(d-d'),$$

$$(12)$$

where $\delta_{uj}\begin{cases}1, & u = j \\ 0, & u \neq j\end{cases}$ and $\phi_{uj}^{(k)}(0) = \begin{cases}1, & u = j \\ 0, & u \neq j\end{cases}$. A patient starting in state u can be in state j on day d either if the patient stays in ward u for d days before transitioning to ward j (the first term of Equation (12)), or $u = j$ and they never left u during the period $[0, d]$ (the second term of Equation (12)), or the patient left u at least once and finally reached j by day d (the third term of Equation (12)). Consequently, $\gamma_j^{(k)}(d)$ can be expressed as sum-product of all possible initial states to ward j (Equation 13).

$$\gamma_j^{(k)}(d) = \sum_{u\in\mathcal{U}}\rho_u^{(k)}\phi_{uj}^{(k)}(d), \qquad (13)$$

γ from Equation (13) becomes an input to the *scheduling* model explained in next section. The semi-Markov

process estimates, $\hat{\Theta}$, can be used for finding the length-of-stay distribution of each patient type as well as the expected mean length of stay in each ward and its variance. Equations to compute these are given in the following subsection as they may be useful for other research objectives or purposes.

3.1.6. Computing Patient Length-of-Stay Distributions. Length-of-stay in a ward (V). For a patient of type k, we estimate the expected days spent by the patient in each ward using the indicator function on the interval transition probability $\Phi^{(k)}$. Let $\bar{V}^{(k)} = [\bar{v}_{uj}^{(k)}]$; $u, j \in \mathcal{U}$; $k \in \mathcal{K}$, where $v_{uj}^{(k)}$ denotes the number of days the patient will spend in j given their initial state was in ward u. The mean of $v_{uj}^{(k)}$ can be computed using Equation (14) given below.

$$\bar{v}_{uj}^{(k)} = \sum_{d=1}^{\infty}\phi_{uj}^{(k)}(d). \qquad (14)$$

The second moment of $v_{uj}^{(k)}$ is given by

$$\bar{v}_{uj}^{2(k)} = \bar{v}_{uj}^{(k)}(2\bar{v}_{uj}^{(k)} - 1). \qquad (15)$$

Thus, the variance of the days spent by a patient in a state can be given by

$$\check{v}_{uj}^{(k)} = \bar{v}_{uj}^{2(k)} - (\bar{v}_{uj}^{(k)})^2 \quad \forall u, j \in \mathcal{U}. \qquad (16)$$

Total hospital length-of-stay (LOS). To get the distribution on LOS for entire hospital stay, we calculate the first-passage-time probabilities, denoted by F. $F^{(k)} = [f_{uj}^{(k)}(d)]$, $u, j \in \mathcal{U}$; $v = 1, 2, \ldots$; $k \in \mathcal{K}$, where $f_{uj}^{(k)}(d)$ is the probability that the first passage from state u to j will take exactly d days for patients of type k. This event can occur if a patient makes a direct transition from u to j on day d, or the patient transitions to any other state l on any day before d and then takes first passage from l to j. The second component is recursive and thus takes into account any number of transitions between any states (except the absorbing state) to reach state j from u in d days.

$$f_{uj}^{(k)}(d) = P_{uj}^{(k)}H_{uj}^{(k)}(d) + \sum_{l\in\underline{\mathcal{U}}\setminus\{j\}}\sum_{d'=1}^{d}P_{ul}^{(k)}H_{ul}^{(k)}(d')f_{lj}^{(k)}(d-d').$$

$$(17)$$

Using $f_{u\bar{u}}^{(k)}$, where $\bar{u} \in \bar{\mathcal{U}}$, and the initial state probability $\boldsymbol{\rho}$, we can get the distribution for LOS. $f_{u\bar{u}}^{(k)}$ denotes the first-passage-probability for a patient's flow from any initial state u to a discharge state $\bar{u}$. If the initial state is unknown then we use Eq. Eq. 18. Otherwise if the initial state is known, say u, then the distribution is given by $f_{u\bar{u}}^{(k)}$ itself.

$$L^{(k)}(d) = \sum_{u \in \underline{\mathcal{U}}} \rho_u^{(k)} f_{u\bar{u}}^{(k)}(d) \quad d = 1, 2, \dots \quad (18)$$

3.1.7. Elective and Emergency Inpatient Census Model.

In this section, we describe how we integrate the semi-Markov stochastic location processes generated from our SMM method with different arrival processes to create a stochastic ward census process. This section, as well as section 3.2, presents an elective scheduling optimization approach focused on hospital patient throughput (i.e., admission volume) and congestion (e.g., bed block, off-ward placement of patients) that is based on the work by Helm and Van Oyen (2014). The purpose of these sections is to provide relevant background for possible applications of our CM method in the hospital census forecasting industry as described by our industry co-author. We use the aforementioned optimization approach as a proof of concept to test the value of our improved CM method and demonstrate how our CM approach integrates seamlessly with existing patient flow optimizations. These sections are, therefore, intentionally brief and not intended to present new research in the area of resource optimization.

There are two broad categories of patients that a hospital serves, elective (EL) and emergency (EM). In developing our census model, we separate the two because in the optimization in section 3.2, emergency arrivals are considered uncontrollable while the scheduled elective arrivals become the primary decision variable. To integrate our SMM clustering and trajectory estimates with the optimization as well as the what-if scenarios of interest to the industry, we run the clustering method on EL and EM patients separately. Hence each stream, EL and EM, will have its own set of patient types, $\mathcal{K}$, with their own trajectories determined by our SMM.

As explained in previous sections (3.1.1–3.1.4), we cluster the EL patients into homogeneous groups with similar trajectories. Trajectory estimates, one for each patient *type* (cluster), are computed using Equations (12) and (13). Combining the EL arrival pattern with the semi-Markov trajectory distributions for each patient type, discussed in section 3.1.5, creates a stochastic census process that can be used to calculate the distribution on patient demand for beds at each ward at any time, t. The exact distribution depends on the arrival process.

For EL admissions, we consider a deterministic arrival process, which, when combined with the semi-Markovian patient trajectories, yields a Poisson-Binomial distribution on bed demand at fixed time point t. The deterministic assumption is an approximation of reality, but has been widely used in the literature due to the fact that elective arrivals are controlled and scheduled in advance. Therefore, it is

(1) theoretically possible to achieve close to a deterministic arrival stream, (2) it is highly beneficial to patient flow for hospital managers to work toward a deterministic elective arrival stream and should be a management priority, (3) deviations from the deterministic arrivals can be incorporated for certain distributions and approximated for others—particularly if the variance of the arrival pattern can be adequately approximated as a linear function of the mean.

We model the arrivals of emergency patients using a non-homogeneous Poisson process that varies by day of week. Combining these Poisson arrivals with the semi-Markov stochastic location processes yields a Poisson-arrival-location model (PALM) of emergency census, (see Massey and Whitt (1993) for more details). One feature of a PALM model is that the distribution on demand for beds in any ward for fixed t follows a Poisson distribution.

Having defined the distribution on demand for beds for emergency and elective patients, we now briefly describe an optimization model from the literature (Helm and Van Oyen, 2014) that is subsequently used to demonstrate the importance of a rigorous patient trajectory estimation procedure. We designed our estimation approach to integrate with optimization and what-if scenarios, with this particular optimization being used as a proof of concept that (1) our method integrates well with current optimization approaches, and (2) our method significantly improves the outcome of the optimization when compared with traditional approaches proposed for use with these types of models.

3.2. Resource Scheduling (RS) MIP Model for Elective Admission Scheduling

The RS model we use as proof of concept integrates both EM and EL census models to capture metrics such as blocking and off-ward placement of patients. The two common objectives from the literature that we focus on are as follows: (1) maximizing the number of elective admissions while constraining congestion metrics and (2) minimizing the congestion (e.g., blocking) while maintaining patient throughput. From a management perspective, the first objective allows for increased revenue, while the second objective provides better access and consequently better outcomes for patients. For ease of reference, we present this optimization model in Appendix C.

This concludes the presentation of our CSI approach. In the next section, we develop a simulation to validate the accuracy of the SMM approach for patient clustering and trajectory estimation and to determine the impact of the SMM on optimal solutions to the MIP model.

Figure 4   Pictorial Representation of Transition Probabilities as a Gray-Scale Heat-Map; With Higher Intensity of Gray for Higher Probability

(a) Data generating transition probabilities   **(b)** Estimated cluster's transition probabilities
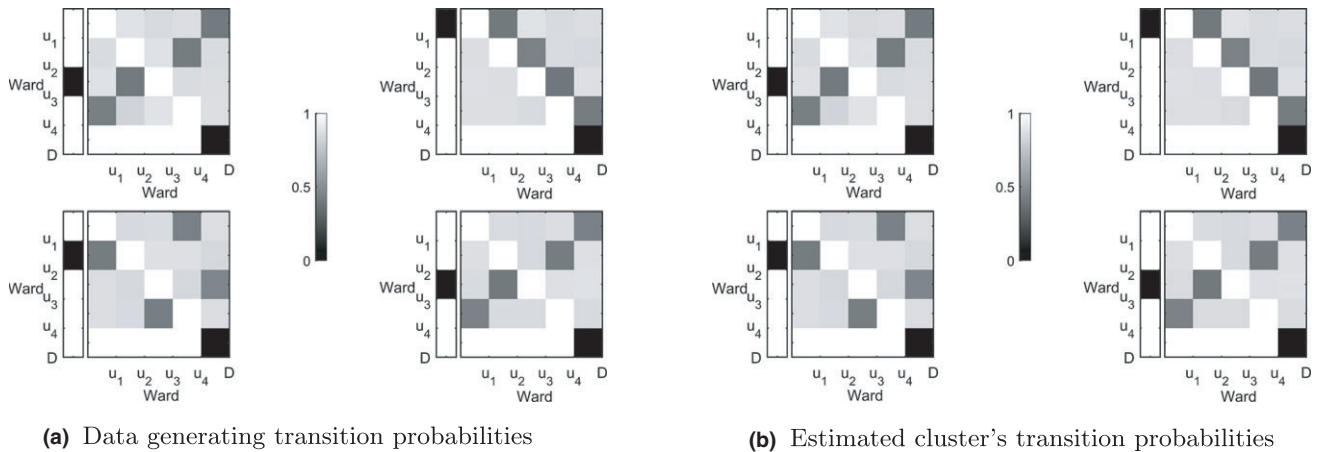
Notes. The heat-map for generating and estimated cluster transition probabilities are shown side-by-side for visual comparison.

4. SMM Validation and Impact on Optimal Scheduling Solutions

In this section, we perform simulation studies to validate the performance of SMM method.

4.1. Evaluating the Accuracy of the SMM Method

We begin with a detailed analysis of the functionality and performance of our SMM method by performing a simulation study of a hospital system with four transient states (wards), $\underline{\mathcal{U}} = \{u_1, \ldots, u_4\}$ and one absorbing state (discharge/death) $\bar{\mathcal{U}} = \{D\}$. We later expand upon this deep-dive to consider a variety of other clustering systems. For the initial simulation, flow sequences for 1000 patients were generated from four different semi-Markov models (corresponding to four different patient types), denoted by $C_s^{(1)}, \ldots, C_s^{(4)}$. As two clusters could be different in $\mathbf{P}$, $\mathbf{H}$, and/or both, we used the following setting that covers all possible scenarios. In the data-generating model, $C_s^{(1)}$ and $C_s^{(2)}$ have different $\mathbf{P}$ but same $\mathbf{H}$, $C_s^{(3)}$ and $C_s^{(4)}$ have same $\mathbf{P}$ and different $\mathbf{H}$, while $C_s^{(2)}$ and $C_s^{(3)}$ have different $\mathbf{P}$ and $\mathbf{H}$. A pictorial representation of the transition probability matrix combined with the initial state probability is shown in Figure 4a. In these plots, the darker the color, the higher the probability. The component mixture weights, π, of the four clusters are $\{0.17, 0.33, 0.25, 0.25\}$, respectively. Additionally, the assignment probabilities in the generating distributions were set less than 0.7 to ensure that the simulation output would be similar to that of a general hospital scenario.

The proposed SMM mixture model was applied to the generated data for various numbers of clusters and the Q function was plotted against the number of clusters, $|\mathcal{K}|$ as shown in Figure 5. As can be seen from the figure, the absolute slope of the Q estimates

significantly drops at $|\mathcal{K}| = 4$ with estimated $\hat{\pi} = \{0.169, 0.332, 0.253, 0.246\}$, which indicates that the true number of clusters and mixture weights were accurately identified by the SMM estimation model. No similar clusters were found by the pairwise hypothesis tests discussed in section 3.1.4. To assess the accuracy of the estimated parameters $\hat{\pi}^{(k)}$, $\hat{\rho}^{(k)}$, $\hat{\mathbf{P}}^{(k)}$, $\hat{\mathbf{H}}^{(k)}$ for each of the estimated clusters, we compared them with the parameters of the data-generating model. The pictorial representation of estimated and true probabilities is shown in Figures 4a and b, respectively. The high degree of similarity between the plots in these two figures implies a highly accurate estimation of initial state and transition probabilities. Additionally, we conducted Chi-square and Kolmogorov–Smirnov tests to verify the equality of estimated and true parameters. The p-values of these tests reported in Table 1 are all greater than 0.05, indicating that the equality of estimated and true parameters (null hypothesis) cannot be rejected, that is, they are *statistically* the same at a 95% confidence level. In summary, all the results show a clear one-to-one

Figure 5   Q Function Estimates Against Number of Clusters for the Simulated Data [Color figure can be viewed at wileyonlinelibrary.com]
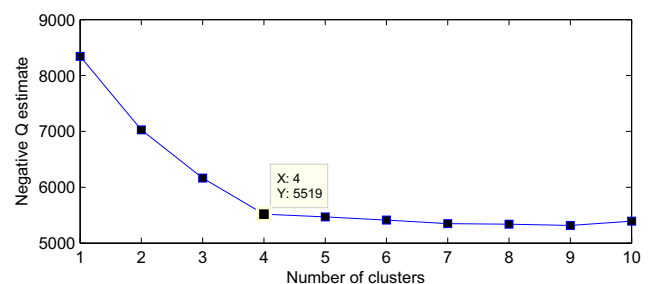
Notes. The improvement in Q estimate becomes insignificant after four clusters.

Table 1 *p*-Values for Matched Cluster Parameters. Higher *p*-Values Compared to the Significance Level Indicates that the Two Compared Distributions were Same

i	*j*	ρ	*P*	*H*
1	2	0.99	0.54	0.99
2	3	0.99	0.17	0.83
3	4	0.99	0.23	0.98
4	1	0.99	0.29	0.99

mapping between estimated and generating (true) cluster parameters, demonstrating the effectiveness of our SMM clustering model at identifying the underlying parameters of the patient flow system.

Next, we provide deeper insight into the functionality of our SMM method by demonstrating the initialization can impact performance and running our method with multiple different random initializations helps overcome this challenge. To do so, we explore three scenarios with 4, 50, and 100 clusters, respectively. The results are reported in Table 2. The table shows the amount of patient data generated in each scenario, and the mean and standard deviation of the length of the simulated patient paths. We remove the data where the path is of length 1 (i.e., the patient arrived to the hospital in a ward for a single time unit and left), since these patients would not be considered hospital inpatients. The sample size remaining is around 75–85% of the original sample (see column 4).

In this table, we show the clustering results from different runs. Each run has a different random initialization. The number of iterations in each run was capped at 50. The f1-score and accuracy are shown as clustering performance measures. The results highlight the differences in the clustering output for different initializations. As expected, the value of objective function corresponds to the accuracy levels—higher the objective value, the higher the accuracy. The

bolded rows of the table indicate the solution that was chosen (out of the five random initializations).

With respect to robustness, we found that the estimates of SMM parameters in each run (in all three scenarios) were found to be statistically similar to the true underlying distributions. This shows that, while the final cluster assignments may be sensitive to initialization, the output of interest, that is, the semi-Markov parameters, are more robust to initialization. As a precaution, however, we suggest multiple random initializations as a means to avoid potentially poor solutions that may be a result of a particular initialization.

Additionally, we show the improvement in objective function and the reduction in cluster reassignment (of trajectories) with each iteration in Figure 6. The figure presents the result for the problem with 50 clusters. The figure shows that (1) the convergence of the algorithm as the iterations progress, and (2) the objective function reaches a upper bound quickly, but the cluster assignments keep changing, although very slightly. The latter observation indicates that multiple cluster solutions gives about the same objective function, although the solution quickly becomes relatively stable. This relates to the identifiability issue and hence our recommendation of setting a maximum on the number of iterations.

4.2. Evaluating the Impact of Patient Flow Estimation on Scheduling Optimization

As discussed in sections 3.1.5 and 3.1.7, the MIP-scheduling model for resource scheduling (RS) uses the estimated trajectory distribution of each patient type as an input. In this section, we study the impact of a better patient path estimation on patient throughput and ward utilization, which are the outcome of RS. To do so, we compare the RS optimization solution using CSI with the true optimal (if the actual
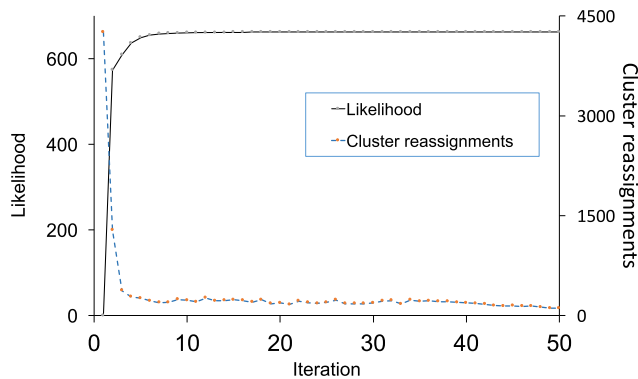
Table 2 Clustering Results under Different Initializations

No of clusters, *K*	No. of patients, *N*	Path lengths' μ, σ	No. of patient paths with length >1	Run	F1-score (%)	Accuracy (%)	Objective value
4	1000	43.2, 32.9	735	1	88.6	91.6	208.82
				2	70.3	81.4	201.28
				3	**88.9**	**91.6**	**209.16**
				4	63.8	78.2	199.8
				5	87.9	90.3	208.26
50	5000	47.5, 49.6	4165	**1**	**90.8**	**92.4**	**620.21**
				2	71.4	85.7	601.09
				3	90.2	91.9	619.82
				4	90.7	92.4	620.02
				5	71.8	85.1	601.11
100	10,000	51.4, 53.2	8504	1	74.6	86.7	1110.57
				2	74.5	86.8	1110.21
				3	88.4	90.3	1121.01
				4	**89.7**	**91.4**	**1121.89**
				5	74.6	86.5	1110.78

Note: The bold row corresponds to the proposed approach.

Figure 6 Improvement in Objective Function and Reduction in Cluster Reassignment as the Algorithm Convergences [Color figure can be viewed at wileyonlinelibrary.com]

distribution were known), and with four other common patient clustering methods: *k*-means clustering, DRG-based clustering, Gaussian clustering, and Markov model-based clustering.

We compare our CSI method with the commonly used attribute-based clustering methods: DRG-based clustering and *k*-means attribute clustering. In these methods, patients are first clustered into groups based on the similarity of their personal and medical attributes. While DRG-based clustering uses the patients' diagnosis (disease) for grouping, *k*-means uses other attributes like age, sex, etc. for a finer grouping. The path distribution for each patient cluster are then estimated empirically. Details are given in Appendices D–E.

There can be drawbacks to *k*-means clustering. Sometimes the data density can be non-convex or severely unbalanced, which affects the method's effectiveness. Hence, we also compare our method with a Gaussian mixture distribution approach for clustering the patients based on their attributes. We also implement and compare our method with a mixture Markov model-based clustering method (Cadez et al. 2000). This Markov model clustering is different from SMM due to its assumption that the holding time distribution is independent of state transition.

The patient path distributions, computed from the patient clusters, are supplied to the MIP model (the *maximum elective admission* formulation presented in section 3.2 and Appendix C). In Table 3, we present

the optimization results for the objectives of maximizing throughput and ward utilization for the same three scenarios studied in the previous section; cluster sizes $K \in \{4, 50, 100\}$.

The table contains a row for "optimal". Here the "optimal" result is drawn by using the known true underlying number of patient types and their path distributions. This result serves as the baseline (or, the upper bound in this case) to assess the performance of other methods.

As shown in Table 3, the outcome from CSI is quite close to the optimal. The Markov model is the next best method, but falls well short of CSI. This is because, similar to the SMM-clustering used in CSI, the Markov model clustering also groups the patients based on similarity in their paths. However, the performance is worse than the one from SMM because it assumes a same holding time distribution, that is, the distribution on length-of-stay in a ward (before moving to another) is always the same. This means that the Markov model is ignoring a critical feature of ward interactions, that is, that holding time and ward transitions are dependent.

The scheduling outcome from empirical patient path distributions (derivation expressions in Appendix E) drawn from attribute-based clustering, viz. *k*-means, DRG, and Gaussian, were significantly poorer than the optimal. Among them, Gaussian performed the worst because of its ineffectiveness in clustering categorical variables present in patient attributes data.

4.3. Applying SMM-Clustering in Practice

In this section, we discuss some of the advantages and disadvantages of the SMM clustering method and what types of problems are best suited for applying our method.

The results of this simulation study show that the proposed CSI yields a schedule in RS that is very near the true optimal, and significantly outperforms existing HASC methods. Our SMM clustering model outperforms traditional attribute-based clustering methods specifically because it takes trajectories into account in the clustering process. Attribute-based

Table 3 Percentage Increase in Service Level Metrics: Throughput from the Number of Elective Patient Admission, and Ward Utilization

	K = 4		K = 50		K = 100	
	Throughput (%)	Utilization (%)	Throughput (%)	Utilization (%)	Throughput (%)	Utilization (%)
(Optimal)	85	49	78	34	89	45
CSI	**81**	**49**	**75**	**33**	**88**	**45**
DRG	21	11	24	14	28	19
k-means	24	24	25	16	32	21
Gaussian	19	9	18	10	21	10
Markov	58	38	51	21	61	37

Note: The bold row corresponds to the proposed approach.

clustering, on the other hand, relies on an indirect relationship between attributes and patient trajectories rather than directly employing trajectories as a clustering approach. This highlights one of the key innovations of our new method. Instructively, our SMM-based clustering method also significantly outperforms the Markov model-based clustering method, which *does* consider patient trajectories. This highlights a second innovation of our SMM method, which is that we properly consider the interaction between wards. That is, our method allows the holding time distribution and ward transitions to be dependent, which is ignored in the Markov model-based clustering method. In addition, SMM clustering effectively differentiates between patients with different lengths-of-stay at the wards. This more subtle modeling difference turns out to have a significant impact on model performance.

As a result, we find that our SMM-clustering approach is most effective being applied to problems with the following characteristics. First, our method performs well in situations where individual characteristics available in the data (e.g., age, sex, and comorbidity) are not adequately explanatory of trajectories. From a patient flow perspective, this feature is highlighted by the example described by Figure 8. Clearly, there are applications outside of patient flow that share this feature.

Second, problems with complex and dependent network interactions can cause simpler methods to perform poorly, creating significant opportunity for our method to outperform existing clustering methods. In particular, non-Markovian networks benefit significantly from relaxing the Markovian assumption in the clustering methodology, as ours does. Non-Markovian networks are quite common in healthcare, since a patient's history has been shown in many contexts to correlate with future requirements and outcomes. However, this feature is not unique to patient flow systems.

Finally, in our context the output of interest is not the clusters themselves, but rather the trajectory distributions derived from the clusters. We have shown that our model is quite robust to local optima and initializations in terms of the overall trajectory distributions because near the optimal EM solution, clusters may continue to change but the overall distributions derived from each cluster remain relatively stable. Our SMM clustering approach provides accurate parameter estimates even if the cluster assignment of all the training data is not the global optimal. This also helps mitigate identifiability concerns, since different many clustering solutions can generate very similar holding time and transition distributions, which is the output of interest. Thus, applications that rely on the semi-Markov distributional output rather than the

actual clustering results themselves are best suited for SMM.

SMM-clustering, however, does rely on Markovian transitions between wards. While this is often not strictly true in patient flow systems, much past literature has found this modeling assumption to be sufficiently accurate. Performance may suffer, however, if transitions are strongly history dependent, thereby making the Markovian assumption a poor representation of reality. In such a situation, the trajectory data can be tested for different orders of the Markovian property and the state space may be able to be expanded to restore the Markovian property if necessary.

Another disadvantage is that, due to large number of parameters to estimate, SMM-clustering requires large amount of data. While, large amounts of data and an underlying Markovian property is common for patient flow problems, other methods such as in Ranjan et al. (2016) can be incorporated to mitigate data scarcity issues.

5. Case Study on Real Hospital Data

In this section, we will study the impact of our integrated framework (CSI) on hospital resource optimization at a partner hospital, and as a holistic tool for the HASC problem. In particular, we focus on validating the trajectory estimation and RS models, as forecasting arrival streams is out of the scope of this study. Hence, we take the arrival stream as given in order to independently evaluate the accuracy and impact of trajectory estimation on the HASC problem.

We use historical data of patient admission and transitions in a hospital with 55 wards including surgical, ICU/CCU, medicine, neurology, oncology, obstetrics, etc. Although, physically the hospital has more than 55 wards, for simplicity several wards were grouped based on expert prior knowledge about their similarity. This system is a good example of a complex hospital system with general ward network structure, transfers, and blocking/congestion.

We obtained one year of data from 2012, with about 11,000 patients who stayed at least one night in the hospital. The dataset includes the patient flow data, length-of-stay at each ward, and patient attribute data, for example, age, sex, diagnosis, etc. The ratio of elective and emergency patients in the data is almost equal. Patients have an average of 4.1 transfers before leaving the hospital. We compare the performance of the CSI model with that of the established clustering and estimation approaches.

We begin with the CM step by applying SMM-based clustering on patient trajectory data to identify patient types. From Figure 7, we can infer that there are 32 patient types. Again, no redundant clusters

were found from pairwise hypothesis testing. The trajectory probability distributions for each of these patient types are computed using Equation (13). Simultaneously conventional partition-based clustering methods, discussed above in section 4.2, viz. *k*-means, DRG, and Gaussian clustering, are used to cluster patients based on the patient's attribute data.

While for DRG, the number of clusters is found from the data (the number of diagnosis types), the criteria for finding the optimal number of clusters with *k*-means and Gaussian are rather subjective. Therefore, in order to have a fair comparison, we use the same number of clusters as chosen by SMM (i.e., 32 clusters). This does not affect the optimization in RS even if we have a few redundant clusters, but prevents the risk of suboptimal results due to under-estimation of the number of clusters. Therefore, the benefits demonstrated by this case study represent a conservative estimate of the true potential benefits when compared to an application to a hospital in the real world. After performing the attribute-based clustering, empirical trajectory distributions are estimated

Figure 7 **The Estimated Q Function against Increasing Number of Clusters for the Real Data in Case Study [Color figure can be viewed at wileyonlinelibrary.com]**

Notes. It is observed that the improvement in Q function is not significant after 32 clusters.

for each patient cluster using the same approach regardless of clustering method.

To verify our claims that two patients with similar attributes may not follow the same trajectory, we observed two patients who were put into the same cluster using the *k*-means; they were both male, aged between 55 and 65 years and were diagnosed for heart disease. Their trajectories within hospital are shown in Figure 8a. In this figure, patient#1 enters the cardiology ward, transitions to the angiography center then to the neurology ward and finally back to cardiology before leaving the hospital. Patient#2, on the other hand, begins their stay in the surgical ward, transitions to the heart clinic, then the ICU, then the operating theater, then to the ICU again and finally back to the surgical ward before being discharged. Although the observed attributes for both patients show similar profiles and a heart disease diagnosis, the trajectories followed by these patients were very different. Observing their trajectories more closely, one can see that patient#2 might have had a severe heart condition, while patient#1 had a relatively milder heart condition only requiring angiography.

When employing our SMM-clustering method, we do not see such dissimilarity in patient trajectories within one cluster. As an example, Figure 8b shows trajectories of a few patients from one of the clusters identified by the SMM approach. Most of the patients in this cluster enter the hospital either in surgical or cardiology wards, then transition to the heart clinic, ICU, operating theater, and finally cardiology before leaving the hospital. There is one case of patient#6 who entered the hospital in ortho and spine center, but then followed similar trajectory of going to heart clinic, ICU, operating theater, and finally cardiology. This could be caused by a heart condition developing during an orthopedic admission, or possibly due to initial off-ward
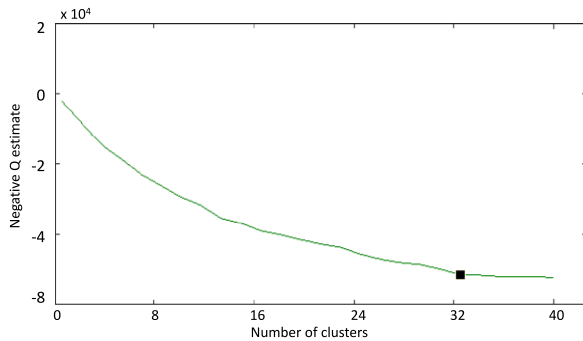
Figure 8 **Trajectories of Patients Belonging to Same Cluster [Color figure can be viewed at wileyonlinelibrary.com]**
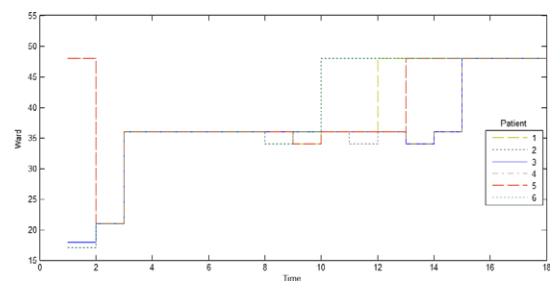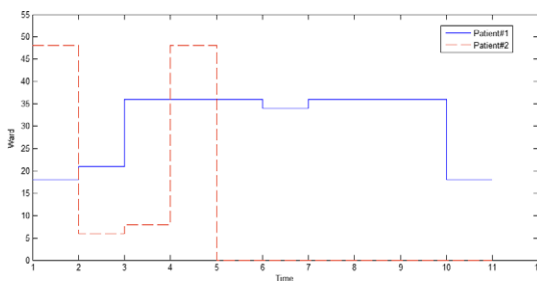
(a) Patient trajectories from a *k*-means cluster   **(b)** Patient trajectories from a SMM based cluster

Notes. It is observed that patients in SMM-based clusters follow more similar trajectories than *k*-means.

Table 4 Comparing the Percentage Improvement in Throughput (Elective Admissions) and Ward Utilization (Workload) for Proposed CSI and other Traditional Methods with Respect to the Current Service and Workload Levels using Real Hospital Data

	Throughput (%)	Utilization (%)
CSI	**97**	**22**
DRG	26	11
k-means	30	8
Gaussian	22	5
Markov	63	19

Note: The bold row corresponds to the proposed approach.

placement (because the cardiology ward was full). It is interesting to see that if we would have used the conventional attribute based clustering this patient would have been put into a orthopedic related cluster, while the SMM approach was able to identify the patient's "true" cluster.

To test the impact of our SMM approach on the RS optimization, we use the *maximum elective admission* formulation given in section 3.2 and Appendix C. The goal is to increase the volume of patients served (throughput), thereby increasing revenues, while maintaining the same level of service and access. The results are shown in Table 4 for the CSI and other methods relative to the baseline current elective admission schedule of the partner hospital.

Our CSI method demonstrates a potential increase in elective admissions of 97%, and an increase of 22% for ward utilization. Similar to the simulation study, the Markov clustering performs second best, with improvements of 63% and 19% for throughput and utilization, respectively. Among the other methods, *k*-means performs the best, yet significantly worse than CSI, with improvement of 30% and 8%. In practice, attribute partition methods like *k*-means and DRG are commonly used, though they leave much to be desired.

This case study of a partner hospital demonstrates the importance of an accurate patient clustering and trajectory estimation method, as using our CSI not only provides a more accurate forecast of the hospital stochastic workload process but also dramatically improves optimization solutions. Furthermore, to the best of our knowledge, our CSI method is the only approach in the extant literature that has all the properties required for effective integration with admission scheduling optimization approaches: *scalable* to hospital of any size, considers *ward interactions*, and accounts for *patient heterogeneity*.

6. Conclusion

The *Hospital Admission Scheduling and Control* problem is comprised of two main components: *census modeling* and *resource scheduling*. Previous work on this long-standing problem has considered one or the other, but not both. In this study, we develop a new method based on *semi-Markov model* (SMM) based clustering for identifying patient type clusters and estimating cluster trajectory distributions that integrates seamlessly with existing scheduling optimization approaches. This integration is proven to be extremely important, as optimal solutions using our SMM approach dramatically outperform optimal solutions using the traditional empirical estimation techniques.

As a theoretical contribution, our novel approach is able to model an entire hospital of any size as a coordinated system with a complex, general network of wards and patient transitions between them. Furthermore, the model has been shown to be *scalable*, accounts for *ward interactions*, and for patient *heterogeneity*, which has not been previously achieved by other methods in the literature. Furthermore, our SMM-clustering is a general purpose algorithm applicable to any movement or sequence data having spatial and temporal dimension, for example, *clickstream* data of users on a website or movement of cell-phone users among a network of towers.

Our SMM approach was designed to integrate with RS approaches that provide an optimal controllable schedule by patient type for each day of week so this approach can be adopted by any specialty or multi-specialty hospital for streamlining their procedures, stabilizing the operating environment for their personnel, improving utilization of hospital resources, and enabling cost savings for both patients and hospitals. The automated, algorithmic approach to clustering and trajectory estimation is also appealing compared to ad-hoc, manual, and heuristic approaches currently employed in practice (which can take months to implement and are difficult to validate statistically).

The SMM-clustering method was validated by simulating data from *known* generating mixture distributions. The SMM estimated clusters and their distributions were found to be statistically the same as the generating mixture distributions at a 95% confidence level. Optimizing the elective schedule based on inputs from our SMM method achieved outcomes that were very close to the "true optimum" (i.e., given perfect knowledge of patient flow dynamics), while the existing traditional method gave performed significantly worse.

A case study using real hospital data showed that the number of elective admissions could be increased by 97% (with the same level of access) compared to only a 30% increase using traditional empirical methods (which are comparable to previous optimization improvements reported in the literature). Moreover,

the average ward utilization could be improved by 22% using our approach compared with only an 8% improvement using the traditional approach.

In conclusion, our approach develops a novel method for spatio-temporal clustering and trajectory estimation that has a profound impact on an important patient flow problem with the potential to improve revenues and/or cost, quality, and access to care.

Acknowledgments

Appendix A. Derivation of SMM-Clustering update Expressions for EM Algorithm

In this Appendix, we present the derivation of parameter update expressions for the EM algorithm in section 3.1.2. As mentioned in the section, we have to obtain the posterior distributions of the parameters to find their optimal estimates that maximizes Equation (9).

We use Dirichlet prior distributions, given in Equation (11), for the parameters. The Dirichlet hyperparameters for parameters in $\boldsymbol{\Theta} = \{\pi^{(k)}, \boldsymbol{\rho}^{(k)}, \mathbf{P}^{(k)}, \mathbf{H}^{(k)}\}, k \in \mathcal{K}$ are denoted by $\{a_\pi^{(k)}, a_\rho^{(k)}, a_P^{(k)}, a_H^{(k)}\}, k \in \mathcal{K}$, respectively. For each model parameter, the hyperparameters can be set to equal values, if there is no specific prior knowledge (non-informative prior). Besides, we assume the parameters are independent. Using it with the conditions on probability sums equal to 1 in Equation (1) and parameter independence assumptions gives the following expressions for prior probabilities,

$$p(\pi) \propto \prod_{k \in \mathcal{K}} \left(\pi^{(k)}\right)^{a_\pi^{(k)}-1}$$

$$p(\boldsymbol{\rho}) \propto \prod_{k \in \mathcal{K}} \prod_{u \in \mathcal{U}} \left(\rho_u^{(k)}\right)^{a_{\rho,u}^{(k)}-1},$$

$$p(\mathbf{P}) \propto \prod_{k \in \mathcal{K}} \prod_{u \in \mathcal{U}} \prod_{j \in \mathcal{U}} \left(P_{uj}^{(k)}\right)^{a_{P,uj}^{(k)}-1}$$

$$p(\mathbf{H}) \propto \prod_{k \in \mathcal{K}} \prod_{u \in \mathcal{U}} \prod_{j \in \mathcal{U}} \prod_{v \in \mathcal{T}} \left(H_{uj}^{(k)}(v)\right)^{a_{H,uj}^{(k)}(v)-1} \quad (A1)$$

Furthermore, using the parameter independence, the prior distribution for $\boldsymbol{\Theta}$ is,

$$p(\boldsymbol{\Theta}) = p(\pi)p(\boldsymbol{\rho})p(\mathbf{P})p(\mathbf{H}). \quad (A2)$$

Plugging Equations (A2) and (2) into Equation (9), and using the hyperparameters mentioned in section 3.1.2, we get,

$$Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(p)})$$

$$= \mathbb{E}_{\boldsymbol{\Theta}^{(p)}}\left[\log(p(\mathbf{Y}|\boldsymbol{\Theta})p(\boldsymbol{\Theta}))\right]$$

$$= \sum_{n=1}^{N} \sum_{k \in \mathcal{K}} \Omega_{nk}(\boldsymbol{\Theta}^{(p)}) \log\left[\pi^{(k)} p_{\boldsymbol{\Theta}}(\mathbf{y}^{(n)}|z^{(n)}=k)\right] + \log p(\boldsymbol{\Theta})$$

$$= \sum_{n=1}^{N} \sum_{k \in \mathcal{K}} \Omega_{nk}(\boldsymbol{\Theta}^{(p)}) \log\left[\pi^{(k)} \rho_{u_1}^{(k)} \prod_{l=1}^{L^{(n)}} \left\{P_{u_l,u_{l+1}}^{(k)} \cdot H_{u_l,u_{l+1}}^{(k)}(v_l)\right\}\right]$$

$$\quad + \log p(\pi)p(\boldsymbol{\rho})p(\mathbf{P})p(\mathbf{H})$$

$$= \sum_{n=1}^{N} \sum_{k \in \mathcal{K}} \log\left[\left(\pi^{(k)}\right)^{\Omega_{nk}(\boldsymbol{\Theta}^{(p)})} \left(\rho_{u_1}^{(k)}\right)^{\Omega_{nk}(\boldsymbol{\Theta}^{(p)})}\right.$$

$$\left.\prod_{l=1}^{L^{(n)}} \left\{\left(P_{u_l,u_{l+1}}^{(k)}\right)^{\Omega_{nk}(\boldsymbol{\Theta}^{(p)})} \cdot \left(H_{u_l,u_{l+1}}^{(k)}(v_l)\right)^{\Omega_{nk}(\boldsymbol{\Theta}^{(p)})}\right\}\right]$$

$$\quad + \log p(\pi)p(\boldsymbol{\rho})p(\mathbf{P})p(\mathbf{H})$$

$$\propto \log\left[\prod_{k \in \mathcal{K}} \left(\pi^{(k)}\right)^{\left(\sum_{n=1}^{N} \Omega_{nk}(\boldsymbol{\Theta}^{(p)})+a_\pi^{(k)}-1\right)}\right]$$

$$+ \sum_{k \in \mathcal{K}} \log\left[\prod_{u \in \mathcal{U}} \left(\rho_u^{(k)}\right)^{\left(\sum_{n=1}^{N} \Omega_{nk}(\boldsymbol{\Theta}^{(p)})\kappa(u_1,u)+a_{\rho,u}^{(k)}-1\right)}\right]$$

$$+ \sum_{k \in \mathcal{K}} \sum_{u \in \mathcal{U}} \log\left[\prod_{j \in \mathcal{U}} \left(P_{uj}^{(k)}\right)^{\left(\sum_{n=1}^{N} \Omega_{nk}(\boldsymbol{\Theta}^{(p)})\bar{\kappa}_{uj}(\mathbf{y}^{(n)})+a_{P,uj}^{(k)}-1\right)}\right]$$

$$+ \sum_{k \in \mathcal{K}} \sum_{u \in \mathcal{U}} \sum_{j \in \mathcal{U}} \log\left[\prod_{v \in \mathcal{T}} \left(H_{uj}^{(k)}(v)\right)^{}\right.$$

$$\left.\cdot \left(\sum_{n=1}^{N} \Omega_{nk}(\boldsymbol{\Theta}^{(p)})\tilde{\kappa}_{uj,v}(\mathbf{y}^{(n)})+a_{H,uj}^{(k)}(v)-1\right)\right]$$

$$\propto \log\left[\pi^{(k)} \sim \text{Dirichlet}\left(\sum_{n=1}^{N} \Omega_{nk}(\boldsymbol{\Theta}^{(p)})+a_\pi^{(k)}\right)\right]$$

$$+ \log\left[\rho_u^{(k)} \sim \text{Dirichlet}\left(\sum_{n=1}^{N} \Omega_{nk}(\boldsymbol{\Theta}^{(p)})\kappa(u_1,u)+a_{\rho,u}^{(k)}\right)\right]$$

$$+ \log\left[P_{uj}^{(k)} \sim \text{Dirichlet}\left(\sum_{n=1}^{N} \Omega_{nk}(\boldsymbol{\Theta}^{(p)})\bar{\kappa}_{uj}(\mathbf{y}^{(n)})+a_{P,uj}^{(k)}\right)\right]$$

$$+ \log\left[H_{uj}^{(k)}(v) \sim \text{Dirichlet}\left(\sum_{n=1}^{N} \Omega_{nk}(\boldsymbol{\Theta}^{(p)})\tilde{\kappa}_{uj,v}(\mathbf{y}^{(n)})\right.\right.$$

$$\left.\left. + a_{H,uj}^{(k)}(v)\right] \quad (A3)$$

where, $\kappa(x,y)$ is an indicator function equal to 1 if $x = y$, $\bar{\kappa}_{uj}(\mathbf{y}^{(n)})$ is the count function equal to the number of times transition was made from state u to j in trajectory $\mathbf{y}^{(n)}$, and $\tilde{\kappa}_{uj,v}(\mathbf{y}^{(n)})$ is the count function equal to the number of times transition was made from state u to j, in trajectory $\mathbf{y}^{(n)}$, when length of stay at state u was v time units.

As shown in Equation (A3), the posteriors of the model parameters are Dirichlet distributions with updated hyperparameters. The posterior of any Dirichlet variable, $x_1, \ldots, x_m \sim \text{Dirichlet}(a_1, \ldots, a_m)$ is

maximized at $E[x_i] = \frac{a_i}{\sum_{i'=1}^{m} a_{i'}}, \forall i$. Thus, the parameter estimates to maximize Equation (9) are,

$$\pi^{(k)(p+1)} = \frac{\sum_{n=1}^{N} \Omega_{nk}(\boldsymbol{\Theta}^{(p)}) + a_{\pi}^{(k)}}{\sum_{k' \in \mathcal{K}} \left[\sum_{n=1}^{N} \Omega_{nk'}(\boldsymbol{\Theta}^{(p)}) + a_{\pi}^{(k')} \right]}, \forall k \in \mathcal{K}.$$

$$\rho_u^{(k)(p+1)} = \frac{\sum_{n=1}^{N} \Omega_{nk}(\boldsymbol{\Theta}^{(p)}) \kappa(u_1, u) + a_{\rho,u}^{(k)}}{\sum_{u' \in \mathcal{U}} \left[\sum_{n=1}^{N} \Omega_{nk}(\boldsymbol{\Theta}^{(p)}) \kappa(u_1, u') + a_{\rho,u'}^{(k)} \right]},$$
$$\forall u \in \mathcal{U}, k \in \mathcal{K}$$

$$P_{uj}^{(k)(p+1)} = \frac{\sum_{n=1}^{N} \Omega_{nk}(\boldsymbol{\Theta}^{(p)}) \bar{\kappa}_{uj}(\mathbf{y}^{(n)}) + a_{P,uj}^{(k)}}{\sum_{j' \in \mathcal{U}} \left[\sum_{n=1}^{N} \Omega_{nk}(\boldsymbol{\Theta}^{(p)}) \bar{\kappa}_{uj'}(\mathbf{y}^{(n)}) + a_{P,uj'}^{(k)} \right]},$$
$$\forall u, j \in \mathcal{U}, k \in \mathcal{K}$$

$$H_{uj}^{(k)}(v)^{(p+1)} = \frac{\sum_{n=1}^{N} \Omega_{nk}(\boldsymbol{\Theta}^{(p)}) \tilde{\kappa}_{uj,v}(\mathbf{y}^{(n)}) + a_{H,uj}^{(k)}(v)}{\sum_{v' \in \mathcal{T}} \left[\sum_{n=1}^{N} \Omega_{nk}(\boldsymbol{\Theta}^{(p)}) \tilde{\kappa}_{uj,v'}(\mathbf{y}^{(n)}) + a_{H,uj}^{(k)}(v') \right]},$$
$$\forall u, j \in \mathcal{U}, v \in \mathcal{T}, k \in \mathcal{K}$$

Appendix B. Alternative Approximate SMM-Clustering Algorithm

An alternate algorithm (Algorithm 2) for SMM-Clustering is proposed here. This is an approximate but faster algorithm than Algorithm 1. Algorithm 2 does a hard cluster assignment as opposed to a soft assignment in the latter. Due to this, a nested loop in function SMMPARAMETERS is not needed. As a result, the computation here reduces by a factor of K. Since this operation is repeated in each iteration, the overall computation time is significantly reduced in this approximate implementation. Besides, the results are found to be close to the results from Algorithm 1. Therefore, if efficiency is critical, this alternate algorithm is recommended.

Algorithm 2. Approximate SMM-Clustering Algorithm

Input: Trajectory data, $\mathbf{Y} = \{\mathbf{y}^{(n)}; n=1,\dots,N\}$, number of clusters, K.
 Initialize:
  $z^{(n)} \leftarrow \text{rand}(1, K), n = 1, \dots, N.$ ▷ s.t. at least one trajectory assigned to each k
  $\Omega \leftarrow \{1/K\}_{N \times K}$
  $a_\pi \leftarrow \varepsilon/K;\ a_\rho \leftarrow \frac{\epsilon}{|\mathcal{U}| \times K};$
  $a_\mathbf{P} \leftarrow \frac{\epsilon}{|\mathcal{U}| \times |\mathcal{U}| \times K};\ a_\mathbf{H} \leftarrow \frac{\epsilon}{|\mathcal{U}| \times |\mathcal{U}| \times |\mathcal{U}| \times K}$ ▷ Prior hyperparameters
 for iter $=1,\dots,$ maxIter **do**
  $\boldsymbol{\Theta} \leftarrow$ SMMPARAMETERS$(\Omega, \mathbf{z})$
  $\Omega \leftarrow$ MEMBERSHIPPROB$(\boldsymbol{\Theta})$
  $z^{(n)} \leftarrow \arg\max_k \Omega_{n,k}, n = 1, \dots, N$
 end for
 function MEMBERSHIP PROB$(\boldsymbol{\Theta})$
  $\Omega \leftarrow \mathbf{0}_{N \times K}$

(continued)

for $k=1,\dots,K$ **do**
  for $n=1,\dots,N$ **do**
   Fetch trajectory sequence, $\mathbf{y}^{(n)} = (\{u_1, v_1\}, \dots, \{u_{L^{(n)}}, v_{L^{(n)}}\}, \{\bar{u}\})$
   $\Omega_{n,k} \leftarrow \pi^{(k)} * \rho_{u_1}^{(k)}$
   for $l = 1, \dots, L^{(n)}$ **do**
    $\Omega_{n,k} \leftarrow \Omega_{n,k} * \mathbf{P}_{u_l,u_{l+1}}^{(k)} * \mathbf{H}_{u_l,u_{l+1}}^{(k)}(v_l)$
   end for
  end for
 end for
 $\Omega_{n,k} \leftarrow \frac{\Omega_{n,k}}{\sum_{k'=1}^{K} \Omega_{n,k'}}; k = 1, \dots, K, n = 1, \dots, N$ ▷ Normalizing for,
 $\sum_{k=1}^{K} \Omega_{n,k} = 1, \forall n = 1, \dots, N$
 return Ω
end function
function SMMPARAMETERS $(\Omega, \mathbf{z})$
 $\pi^{(k)} \leftarrow a_\pi + \sum_{n=1}^{N} \mathbf{1}(z^{(n)} = k);\ k = 1, \dots, K$
 $\pi^{(k)} \leftarrow \pi^{(k)} / \sum_{k'=1}^{K} \pi_{k'}$ ▷ Normalizing
 $\rho \leftarrow \mathbf{0}_{\mathcal{U} \times K}; \mathbf{P} \leftarrow \mathbf{0}_{\mathcal{U} \times \mathcal{U} \times K}; \mathbf{H} \leftarrow \mathbf{0}_{\mathcal{U} \times \mathcal{U} \times \mathcal{T} \times K}$
 $\rho_u^{(k)} \leftarrow a_\rho; u \in \underline{\mathcal{U}}, k = 1, \dots, K$
 $\mathbf{P}_{u,u'}^{(k)} \leftarrow a_\mathbf{P}; u, u' \in \underline{\mathcal{U}}, u \neq u', k = 1, \dots, K$
 $\mathbf{H}_{u,u'}^{(k)}(v) \leftarrow a_\mathbf{H}; u, u' \in \underline{\mathcal{U}}, u \neq u', v \in \mathcal{T}, k = 1, \dots, K$
 for $n=1,\dots,N$ **do**
  Fetch trajectory sequence, $\mathbf{y}^{(n)} = (\{u_1, v_1\}, \dots, \{u_{L^{(n)}}, v_{L^{(n)}}\}, \{\bar{u}\})$
  $\rho_{u_1}^{z^{(n)}} \leftarrow \rho_{u_1}^{z^{(n)}} + 1$
  for $l = 1, \dots, L^{(n)}$ **do**
   $\mathbf{P}_{u_l,u_{l+1}}^{z^{(n)}} \leftarrow \mathbf{P}_{u_l,u_{l+1}}^{z^{(n)}} + 1$
   $\mathbf{H}_{u_l,u_{l+1}}^{z^{(n)}}(v_l) \leftarrow \mathbf{H}_{u_l,u_{l+1}}^{z^{(n)}}(v_l) + 1$
  end for
 end for ▷ Normalizing as per Eq. 1
 for $k=1,\dots,K$ **do**
  $\rho_u^{(k)} \leftarrow \rho_u^{(k)} / \sum_{u' \in \mathcal{U}} \rho_{u'}^{(k)}; \forall u \in \mathcal{U}, k = 1, \dots, K$
  $\mathbf{P}_{u,j}^{(k)} \leftarrow \mathbf{P}_{u,j}^{(k)} / \sum_{j' \in \mathcal{U}} \mathbf{P}_{u,j'}^{(k)}; \forall u, j \in \mathcal{U}, k = 1, \dots, K$
  $\mathbf{H}_{u,j}^{(k)}(v) \leftarrow \mathbf{H}_{u,j}^{(k)}(v) / \sum_{v' \in \mathcal{T}} \mathbf{H}_{u,j}^{(k)}(v'); \forall u, j \in \mathcal{U}, v \in \mathcal{T}, k = 1, \dots, K$
 end for
 return $\boldsymbol{\Theta} = \{\pi^{(k)}, \rho^{(k)}, \mathbf{P}^{(k)}, \mathbf{H}^{(k)}; k=1, \dots, K\}$
end function
Output: $\hat{\boldsymbol{\Theta}} = \{\hat{\Theta}^{(k)}\}, k = 1, \dots, K$; and hard cluster assignments $\mathbf{z}$.

Appendix C. Elective Scheduling Optimization MIP Formulation

In this Appendix we present, an optimization model from the literature (Helm and Van Oyen, 2014) that is used to demonstrate the importance of a rigorous patient trajectory estimation procedure. We designed our estimation approach to integrate with optimization and what-if scenarios, with this particular optimization being used as a proof of concept that (1) our method integrates well with current optimization approaches and (2) our method significantly improves the outcome of the optimization when compared with traditional approaches proposed for use with these types of models. We begin by describing the model parameters and then present the optimization model with brief description of the objective and constraints. For a more detailed description of the optimization approach, we refer the readers to Helm and Van Oyen (2014).

Sets

$\mathcal{K}$	set of all patient types
$\mathcal{U}$	set of hospital wards

Hospital parameters

ζ	vector of ward capacities
η	vector of total cancelations attributed for each ward
b	limit on the average number of blockages per week
$\mathbf{o}$	vector of limit on the average number of off-unit patients allowed for each ward
$\mu_d^{(k)}$	current elective admission volume of type k patients on day d
$\bar{\mu}_d^{(k)}$	maximum number of elective admissions of type k allowed on day d
$\mathbf{R}$	reward vector where R_k is the reward for admitting patient oftype k

Patient trajectory and census distributions

$\gamma_u^{(k)}(d_1)$	probability that an elective patient of type k requires a bed inward u, d_1 days after admission (trajectory distribution)
$p_{u,d}(n)$	probability that there are n emergency patients demanding a bedin ward u on day d
$\bar{p}_d(n)$	probability that there are n emergency patients demanding a bedin the hospital on day d

Decision Variables

$\Psi_d^{(k)}$	number of type $k \in \mathcal{K}$ patients scheduled on day d
$\delta_{d,n}$	number of blockages if there are n emergency patients in the hospital on day d
$\acute{o}_{d,n}^u$	number of ward u off-unit patients on day d if there are n emergency patients in ward u

The patient trajectory and census distribution parameters are computed offline as explained earlier in this section. Since the PALM model for emergency patient bed demand is exogenous to the decision variable, this too is calculated off-line, with the results captured as $p_{u,d}(n)$ and $\bar{p}_d(n)$. We consider a weekly planning horizon that repeats itself every week, generating a cyclostationary system that varies by day of week. The objective is to maximize the throughput of the sum of elective patient admissions (over the planning horizon) of each type weighted by a "reward" vector $\mathbf{R}$ ($\mathbf{1}$ denotes a column vector of all ones). The reward vector gives flexibility to allow the model to treat one patient type differently from another, for example, the model can prioritize one patient type over another with respect to patient criticality, projected revenue generated by the admission, or other strategic priority. The formulation is as follows:

$$\max_{\Theta,\delta,\hat{\delta}} \mathbf{R} \cdot \Psi \cdot \mathbf{1}, \qquad (C1)$$

s.t.

$$\delta_{d_1,n} \geq n - \sum_{u \in \mathcal{U}} \left(\zeta_u - \sum_{d_2=1}^{7} \sum_{k \in \mathcal{K}} \Psi_{d_2}^{(k)} \cdot \sum_{n'=0}^{\infty} \gamma_u^{(k)}(7n' + d_1 - d_2) \right),$$
$$d_1 = 1, \ldots, 7; n = 1, 2, \ldots \qquad (C2)$$

$$\sum_{d=1}^{7} \sum_{n=0}^{\infty} \bar{p}_d(n)\delta_{d,n} \leq b, \qquad (C3)$$

$$\delta_{d,n+1} \geq \delta_{d,n} \quad d = 1, \ldots, 7; n = 1, 2, \ldots, \qquad (C4)$$

$$\acute{o}_{d_1,n}^u \geq n + \sum_{d_2=1}^{7} \sum_{k \in \mathcal{K}} \Psi_{d_2}^{(k)} \cdot \sum_{n'=0}^{\infty} \gamma_u^{(k)}(7n' + d_1 - d_2)$$
$$- \zeta_u - \eta_u \sum_{d=0}^{7} \sum_{n'=0}^{\infty} \delta_{d,n'} \cdot \bar{p}_d(n') \qquad , \quad (C5)$$
$$\forall u \in \mathcal{U}; d_1 = 1, \ldots, 7; n = 1, 2, \ldots$$

$$\sum_{n=0}^{\infty} p_{u,d}(n)\acute{o}_{d,n}^u \leq \mathbf{o}_u \quad \forall u \in \mathcal{U}; d = 1, \ldots, 7, \qquad (C6)$$

$$\acute{o}_{d,n+1}^u \geq \acute{o}_{d,n}^u \quad d = 1, \ldots, 7; n = 1, 2, \ldots, \qquad (C7)$$

$$\sum_{d=1}^{7} \Psi_d^{(k)} \geq \sum_{d=1}^{7} \mu_d^{(k)} \quad \forall k \in \mathcal{K}, \qquad (C8)$$

$$\Psi_d^{(k)} \leq \bar{\mu}_d^{(k)} \quad \forall k \in \mathcal{K}; d = 1, \ldots, 7 \qquad (C9)$$

$$\Psi_d^{(k)}, \delta_{d,n}, \acute{o}_{d,n}^u \in \mathbb{Z}^+$$

The constraints of this model are primarily for constraining the blockages faced by the patients, limiting off-ward placement, and respecting the hospital resource limits. Since the purpose of this work was to demonstrate how CM can be improved by developing methods that integrate with optimization, and not to provide new optimization methods, we briefly describe the optimization presented here. Greater detail regarding this approach can be found in Helm and Van Oyen (2014). Constraints (C2) calculate the number of blocked patients at the hospital level if n emergency patients are in the hospital on day d_1. This sets the helper variable, $\delta_{d,n}$ which is subsequently used to calculate expected blockages according to the distribution on the emergency patient bed demand stochastic process in the left-hand side (LHS) of Constraints (C3) by multiplying the indicator of whether the nth patient would be blocked by the probability of seeing n emergency patients in the hospital. The right-hand side constrains the expected blocked patients to be less than some target level, b, which can be chosen by management. Constraint (C4) is a cut that is added to the formulation that significantly improves model solution speed.

Similar to the constraints (C2–C4) for blockages, we have constraints in C5–C7 for approximating and limiting expected off-unit census. An additional term in Equation (C5), $\eta_u \Sigma_{d=0}^{7} \Sigma_{n'=0}^{\infty} \delta_{d,n'} \cdot \bar{p}_d(n')$, subtracted from the otherwise expected number of off-unit census gives patients who were blocked and not able to be admitted to the hospital in the first place.

Constraints in (C7) ensures that the proper mix of patients is respected. Specifically, it ensures that each patient type has at least as many admissions each week as they did prior to optimization. Constraints (C9) ensure that the model respects the hospital resource capacity for a day. For example, hospitals frequently avoid admitting elective patients on Sundays, which could be achieved by setting $\bar{\mu}_{Sunday}^{(k)} = 0$.

Appendix D. Assigning Attributes to Patients in Simulation Study

Here we elaborate on synthesis patient attributes for simulation study in section 4. For brevity, we show it for $K = 4$. In the data generation step for this problem, after patient trajectories were generated from four semi-Markov processes, three attributes, viz. age, gender, and diagnosis (with three diagnoses being D1, D2, and D3), were assigned to the patients such that any attribute triplet has the possibility of being in any cluster; for example, a 30-year-old female with diagnosis D1 could potentially be from any of the four clusters. This resembles real-world challenges involved in patient trajectory estimation by simulating the fact that two patients with the same attributes may have different trajectories; that is, the attributes are not adequately capturing patient heterogeneity. In practice, patient attributes are capable of capturing some of the patient heterogeneity so we ensure that clusters contain patients whose attributes are mostly

similar by adhering to a near-Pareto principle (see the three attribute generating tables in Table D1). That is, clusters are composed mostly of similar patient attributes with a mix of patients who have different attributes. This distribution of attributes is designed to be fair to the traditional approach and capture the reality that attributes do have differentiating power, but cannot completely specify a patients likely trajectory.

In section 6, we assign physical attributes to patient for our simulation study. We perform a conservative assignment, in favor of traditional patient clustering method, by giving higher chance of patients within a *true* cluster having similar attributes. Table D1 below shows the generating distributions for the patient attributes within each cluster. As shown in the table, age is taken from a normal distribution with different means (Table D1a), and sex and diagnosis (Table D1b–D1c) are taken according to a Bernoulli random variable with different success probabilities. The distribution parameters are chosen such that there is high attribute similarity (dissimilarity) between patients within (between) clusters.

Appendix E. Empirical stimation of Patient Trajectories

Once the clusters have been formed using k-means clustering, the trajectory distribution is computed for each cluster independently by normalizing the frequency of transitions of patients between wards as follows:

1. $\rho_u^{(k)} = \frac{\Sigma_{n=1}^N \kappa(\mathbf{y}_1^{(n)}, u)}{\Sigma_{u' \in \mathcal{U}} [\Sigma_{n=1}^N \kappa(\mathbf{y}_1^{(n)}, u')]}$ for $k \in \mathcal{K}$ and $u \in \mathcal{U}$.

2. $P_{uj}^{(k)} = \frac{\Sigma_{n=1}^N \bar{\kappa}_{uj}(\mathbf{y}^{(n)})}{\Sigma_{j' \in \mathcal{U}} [\Sigma_{n=1}^N \bar{\kappa}_{uj'}(\mathbf{y}^{(n)})]}$ for $k \in \mathcal{K}; u \in \mathcal{U}$ and $j \in \mathcal{U}$.

3. $H_{uj}^{(k)}(v) = \frac{\Sigma_{n=1}^N \tilde{\kappa}_{uj,v}(\mathbf{y}^{(n)})}{\Sigma_{v' \in \mathcal{T}} [\Sigma_{n=1}^N \tilde{\kappa}_{uj,v'}(\mathbf{y}^{(n)})]}$ for $k \in \mathcal{K}, u \in \mathcal{U}, j \in \mathcal{U}$ and $v \in \mathcal{T}$.

Table D1 Generating Distributions for Patient Attributes Within *True* Clusters

Cluster	Age
(a) Normal distribution for patient age	
1	$N(20, 3)$
2	$N(30, 3)$
3	$N(40, 3)$
4	$N(50, 3)$

	Sex (%)	
Cluster	M	F
(b) Uniform distribution for patient sex within clusters		
1	80	20
2	20	80
3	70	30
4	30	70

	Diagnosis		
Cluster (%)	D1	D2	D3
(c) Uniform distribution for patient diagnosis			
1	70	20	10
2	20	70	10
3	10	20	70
4	80	10	10

References

Abraham, G., G. B. Byrnes, C. A. Bain. 2009. Short-term forecasting of emergency inpatient flow. *IEEE Trans. Inf. Technol. Biomed.* **13**(3): 380–388.

Adan, I., J. Bekkers, N. Dellaert, J. Vissers, X. Yu. 2009. Patient mix optimisation and stochastic resource requirements: A case study in cardiothoracic surgery planning. *Health Care Manag. Sci.* **12**(2): 129–141.

Aiken, L. H., S. P. Clarke, D. M. Sloane, J. Sochalski, J. H. Silber. 2002. Hospital nurse staffing and patient mortality, nurse burnout, and job dissatisfaction. *JAMA* **288**(16): 1987–1993.

Armony, M., S. Israelit, A. Mandelbaum, Y. N. Marmor, Y. Tseytlin, G. B. Yom-Tov. 2015. On patient flow in hospitals: A data-based queueing-science perspective. an extended version (ev). Technical report, Working paper, Available at http://ie.technion.ac.il/serveng/References/Patient%20flow%20main.pdf.

Bekker, R., P. M. Koeleman. 2011. Scheduling admissions and reducing variability in bed demand. *Health Care Manag. Sci.* **14**(3): 237–249.

Billingsley, P. 1960. *Statistical Inference for Markov Processes*, University of Chicago Press, Chicago, IL.

Billingsley, P. 1961. Statistical methods in markov chains. *Ann. Math. Stati.* **32** (1): 12–40.

Cadez, I., D. Heckerman, C. Meek, P. Smyth, S. White. 2000. Visualization of navigation patterns on a Web site using model-based clustering. *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 280–284, August 20-23, 2000, Boston, MA, USA. https://doi.org/10.1145/347090.347151.

Earnest, A., M. I. Chen, D. Ng, L. Y. Sin. 2005. Using autoregressive integrated moving average (arima) models to predict and monitor the number of beds occupied during a sars outbreak in a tertiary hospital in singapore. *BMC Health Serv. Res.* **5**(1): 1.

Faddy, M., S. McClean. 1999. Analysing data on lengths of stay of hospital patients using phase-type distributions. *Appl. Stoch. Models Bus. Indus.* **15**(4): 311–317.

Fetter, R. B., Y. Shin, J. L. Freeman, R. F. Averill, J. D. Thompson. 1980. Case mix definition by diagnosis-related groups. *Medical Care* **18**(2): i–53.

Green, L. V. 2006. Queueing analysis in healthcare. In R. W. Hall, ed. *International Series in Operations Research & Management Science*. **91**: 281–307.

Griffith, J. R., W. M. Hancock, F. C. Munson. 1976. *Cost Control in Hospitals*. Health Administration Press, Ann Arbor, MI.

Griffin, J., S. Xia, S. Peng, P. Keskinocak. 2012. Improving patient flow in an obstetric unit. *Health Care Manag. Sci.* **15**(1): 1–14.

Hancock, W. M., P. F. Walter. 1979. The use of computer simulation to develop hospital systems. *ACM SIGSIM Simul. Digest* **10**(4): 28–32.

Hancock, W. M., P. F. Walter. 1983. *The "ASCS": Inpatient Admission Scheduling and Control System*. Health Administration Press, Ann Arbor, MI, Chapter IX.

Harper, P. R. 2005. A review and comparison of classification algorithms for medical decision making. *Health Policy* **71**(3): 315–331.

Harper, P. R., A. Shahani. 2002. Modelling for the planning and management of bed capacities in hospitals. *J. Oper. Res. Soc.* **53**(1): 11–18.

Helm, J. E., M. P. Van Oyen. 2014. Design and optimization methods for elective hospital admissions. *Oper. Res.*, **62**(6): 1265–1282.

Irvine, V., S. McClean, P. Millard. 1994. Stochastic models for geriatric in-patient behaviour. *Math. Med. Biol.* **11**(3): 207–216.

Jacobson, S. H., S. N. Hall, J. R. Swisher. 2006. Discrete-event simulation of health care systems. R. W. Hall, ed. *Patient Flow: Reducing Delay in Healthcare Delivery*. Springer, New York, 211–252.

Jones, S. A., M. P. Joy, J. Pearson. 2002. Forecasting demand of emergency care. *Health Care Manag. Sci.* **5**(4): 297–305.

Kao, E. P. 1972. A semi-markov model to predict recovery progress of coronary patients. *Health Serv. Res.* **7**(3): 191.

Kao, E. P. 1974. Modeling the movement of coronary patients within a hospital by semi-Markov processes. *Oper. Res.* **22**(4): 683–699.

Konrad, R., K. DeSotto, A. Grocela, P. McAuley, J. Wang, J. Lyons, M. Bruin. 2013. Modeling the impact of changing patient flow processes in an emergency department: Insights from a computer simulation study. *Oper. Res. Health Care* **2**(4): 66–74.

Littig, S. J., M. W. Isken. 2007. Short term hospital occupancy prediction. *Health Care Manag. Sci.* **10**(1): 47–66.

Marshall, A., S. McClean. 2003. Conditional phase-type distributions for modelling patient length of stay in hospital. *Int. Trans. Oper. Res.* **10**(6): 565–576.

Massey, W. A., W. Whitt. 1993. Networks of infinite-server queues with nonstationary poisson input. *Queueing Syst.* **13**(1–3): 183–250.

Ranjan, C., S. Ebrahimi, K. Paynabar. 2016. Sequence graph transform (sgt): A feature extraction function for sequence data mining. *arXiv preprint arXiv:1608.03533*.

Taylor, G., S. McClean, P. Millard. 2000. Stochastic models of geriatric patient bed occupancy behaviour. *J. R. Statistic. Soc. A* **163**(1): 39–48.

Thomas, W. H. 1968. A model for predicting recovery progress of coronary patients. *Health Serv. Res.* **3**(3): 185.

Weiss, E. N., M. A. Cohen, J. C. Hershey. 1982. An iterative estimation and validation procedure for specification of semi-markov models with application to hospital patient flow. *Oper. Res.* **30**(6): 1082–1104.

Wu, C. J. 1983. On the convergence properties of the em algorithm. *Ann. Stat.* **11**: 95–103.

Zeltyn, S., Y. N. Marmor, A. Mandelbaum, B. Carmeli, O. Greenshpan, Y. Mesika, S. Wasserkrug, P. Vortman, A. Shtub, T. Lauterman, et al. 2011. Simulation-based models of emergency departments: Operational, tactical, and strategic staffing. *ACM Trans. Model. Comput. Simul.* **21**(4): 24.

Zhang, B., P. Murali, M. Dessouky, D. Belson. 2009. A mixed integer programming approach for allocating operating room capacity. *J. Oper. Res. Soc.* **60**(5): 663–673.