Data Driven Modeling, Monitoring and Control for Smart and Connected Systems

by

Chao Wang

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Industrial and Systems Engineering)

at the

UNIVERSITY OF WISCONSIN–MADISON

2019

Date of final oral examination: 06/12/2019

This dissertation is approved by the following members of the Final Oral Committee:
Shiyu Zhou, Professor, Department of Industrial and Systems Engineering
Jeffrey Linderoth, Professor, Department of Industrial and Systems Engineering
Kaibo Liu, Associate Professor, Department of Industrial and Systems Engineering
Kam-Wah Tsui, Professor, Department of Statistics
Xiaojin Zhu, Professor, Department of Computer Sciences

ACKNOWLEDGMENTS

The research presented in this dissertation benefited from valuable insights and support of many people. It is my great pleasure to express my sincere gratitude to all of them.

First and foremost, I would like to thank my PhD advisor, Prof. Shiyu Zhou for his outstanding guidance, mentorship and full support in my PhD studies. Words cannot express my heartfelt gratitude for his efforts towards my development as a good researcher and a future academic mentor. I have been so lucky to benefit from not only his expertise in academic research, but also his great personality in all other aspects.

I would like to express my special thanks to my doctoral committee members Prof. Jeffrey Linderoth, Prof. Kaibo Liu, Prof. Kam-Wah Tsui and Prof. Xiaojin Zhu for serving on my committee and providing suggestions and helps to improve my research.

My thanks also go to my friends in my lab: Junbo Son, Yuhang Liu, Raed Kontar, Salman Jahani, Akash Deep, Jaesung Lee and Congfang Huang. Thank you for being supportive and accompanying me in my PhD studies. You really make my life in Madison enjoyable and memorable.

Lastly and most importantly, I want to thank my parents for their unconditional love, support and encouragement. Without their support, I would not have made this achievement in my life. I would also thank my wife Yuting Lu for her patience, love and company during my study.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Information revolution is turning modern engineering systems into smart and connected systems. The smart and connected systems are defined by three characteristics: tangible physical components that comprise the system, connectivity among components that enables data acquisition and sharing, and smart data analytics and decision making capability. Examples of smart and connected systems include GM's OnStar® tele-service system and the InSite® tele-monitoring system from GE. The unprecedented data availability in smart and connected systems provides significant opportunities for data analytics. For example, since we have observations from potentially a very large number of similar units, we can compare their operations, share the information, and extract some common knowledge to enable accurate prediction and control at the individual level. In addition, for a complex system such as multistage manufacturing processes, we can collect synchronized data from multiple stations within the system so that we can identify the operational relationships among these stations. Such relationship can enable better process control.

On the other hand, the tremendous data volume and types also reveal critical challenges. First, the high dimensional data with heterogeneity often poses difficulties in sharing common information within/across similar units/processes in the smart and connected systems. This problem becomes more severe when the system under the start-up period, where insufficient data and experience could result in the deficiency of data driven approaches. Second, the non-Gaussian data and non-linear relationship among various units impede the quantitative description of the inter-relationship of processes in the smart and connected systems. Although existing non-parametric methods, e.g., Kriging, can deal with these situations to some extent, limited description power (focus on mean value prediction) and lack of physical interpretation are the common drawbacks in these methods. Moreover, the real time monitoring and control for the smart and connected

systems require efficient and scalability algorithms and strategies to meet the rapid and large scale response under advanced sensing and data acquisition environment. Lastly, the efficient control of the smart and connected systems also becomes challenging due to the complex relationship among units. Data-driven methods are required to meet the exigent demands for effectively formulating and solving the control problem.

To address the issues listed above, four tasks are investigated in this dissertation under different applications in the smart and connected systems.

1 Transfer learning among heterogeneous multistage manufacturing processes. A series of data analytical methods for modeling and learning inter-relationships among product quality characteristics in multistage connected manufacturing processes are developed. The methods offer a rigorous way to reveal commonalities among heterogeneous data from different manufacturing processes to benefit the learning in complex connected manufacturing processes.

2 Statistical modeling and inference for Key Performance Indicators (KPI) in production systems. A surrogate model for inference and prediction at distribution level of different KPIs is developed. This model utilizes the pair-copula construction to capture the non-linear association in the non-Gaussian data.

3 Real time contamination detection in water distribution network. A contamination source identification framework is proposed for real time tracking and detection of contamination released in the urban water distribution network. The framework utilizes the Bayesian theory to sequentially update the posterior probability for determining the contamination source upon very limited sensor readings.

4 Control of KPIs in manufacturing production systems. The KPI control problem is formulated as a stochastic optimization problem, where the noise distribution in the cost function depends on the decision variables. The standard uniform distributions are employed to link the KPI relationship surrogate model and the objective function to efficiently solve the KPI control problem.

The proposed methods can be applied to a broad range of data analytics problems, and the emerging challenges in modeling, monitoring and control of smart and connected systems can be effectively addressed.

1 INTRODUCTION

1.1 Motivation

Smart and connected systems and internet of things (IoT), as emerging engineering technologies, have the great potential of bringing broad disruptive societal impacts, particularly on economic competitiveness, quality of life, public health, and essential infrastructure. The smart and connected systems are defined by three characteristics: tangible physical components that comprise the system, connectivity among components that enables data acquisition and sharing, and smart data analytics and decision making capability [1]. Such systems have become increasingly available in practice. Examples include GM's OnStar® tele-service system, the Konnect® system from Kohler, and Connected Enterprise initiative by Rockwell. Various industries have seen the great potential of IoT enabled smart and connected systems and start to invest heavily in this area. Cisco Systems estimates that, by 2022, the potential value at stake from IoT is $14.4 Trillion (representing increased revenues and cost reductions), with the industrial and manufacturing sector having the most potential impact [2]. For example, by utilizing smart and connected system technology in manufacturing systems, we can make system operations transparent and enable smart operation decision to improve various key performance measures and hence competitiveness.

The essential feature of engineering analytics problem for IoT enabled smart and connected system is that a large amount of data from multiple similar units and from multiple components within the system during their operations in real time are available. The unprecedented data availability in smart and connected systems poses both intellectual opportunities and challenges for data analytics and decision making for operations management. The opportunities are: i) In a smart and connected system, we can

often collect data from a large number of similar units. Such data availability provides convenience for knowledge transfer and information fusion among different yet similar units to achieve improved modeling and prediction for a specific individual unit. ii) In a smart and connected system, data are often available in real time from cascaded units within systems. This provides great opportunity for identifying the interactions among the operations of connected units, which will significantly benefit the global decision making and management of the smart and connected system. iii) The tremendous data availability provides remarkable opportunities to rigorously test and validate the intuitive hypothesis and engineering knowledge that often held by field engineers in an un-structured form. On the other hand, the integration of domain knowledge with data-driven analytics can significantly improve model effectiveness and accuracy. However, the intellectual challenges also exist: i) The high dimensional data with heterogeneity often poses difficulties in sharing common information within/across similar units/processes in the smart and connected systems. This problem becomes more severe when the system under the start-up period, where insufficient data and experience could result in the deficiency of data driven approaches. ii) The non-Gaussian data and non-linear relationship among various units impede the quantitative description of the inter-relationship of processes in the smart and connected systems. Although existing non-parametric methods, e.g., Kriging, can deal with these situations to some extent, limited description power (focus on mean value prediction) and lack of physical interpretation are the common drawbacks in these methods. iii) The real time monitoring and control for the smart and connected systems require efficient and scalability algorithms and strategies to meet the rapid and large scale response under advanced sensing and data acquisition environment. iv) The efficient control of the smart and connected systems also becomes challenging due to the complex relationship among units. Moreover, data-driven methods are required to meet the exigent demands

for effectively formulating and solving the control problem.

As a result, novel data driven methods with efficient involvement of domain knowledge play a significant role in fulfilling the promise of smart and connected systems and facilitating the transformation from *data-rich* into *decision-smart*.

1.2 Research Objectives

To address the challenges mentioned in Section 1.1, a series of data-driven statistical methods are proposed in this dissertation that are tailored for the opportunities and the needs of modeling, monitoring and control for the smart and connected systems. More specifically, the research tasks can be summarized as follows

1 Information sharing and transfer learning among individualized smart and connected systems. The interactions among units in the smart and connected systems are crucial to understand and further model the system. For example, the interactions among stations in the multistage manufacturing system reveal the uncertainty flow and manufacturing sequence in the whole process. However, the interaction fitting is very challenging due to the complexity of the system, especially when the data is insufficient in some start-up processes. In this task, we propose to use transfer learning to take advantage of the historical data from existing processes to benefit the understanding of interactions in new process. We use various multistage manufacturing processes as examples to demonstrate the method and deal with heterogeneity problem among different stations in different processes.

2 Inter-relationship modeling and inference with non-linear and non-Gaussian

data. The Normal/Gaussian noise assumption is widely used in engineering applications due to its concise form and practical properties [3]. However, in the production system, especially the queuing network, the Gaussian distribution cannot cover most of the practical situations. Meanwhile, due to the complexity of production system, it is very common that non-linear relationship exist among variables in the production system. As a result, it is very difficult to have close form characterization of the general production system. Our goal in this task is to develop a surrogate model that can quantitatively represent the inter-relationship among variables in the production system.

3 Real time detection for smart and connected systems. The efficiency of anomaly detection becomes significant in the smart and connected system under the unprecedented data availability. The explosive data need to be handled in real time to provide useful information for anomaly detection during the system operation. On the other hand, the large amount of collected data reveals great opportunities on exploring system patterns and rules offline. It is imperative to develop methods to structure, analyze and finally combine such off line information into the online monitoring to benefit the efficiency of anomaly detection.

4 Efficient data-driven control on complex systems. In practice, we often need to control the system performance to be at certain level to optimize the operation schedules and maximize the profit. To achieve this goal, a specific control problem needs to be formulated based on the system variables relationship. However, the complicated variables relationship sets difficulties in formulating the control problem since it is very challenging

to coordinate all the variables to contribute to the desirable system performance under complex variable relationship. We propose to formulate the system control as an endogenous stochastic optimization problem, which is further analyzed, simplified and solved by ordinary stochastic approximation techniques.

Targeting the above mentioned research goals, four specific research tasks that correspond to each chapter of this dissertation are summarized in the next section. The details of state of art and literature survey for each topic are presented in the corresponding chapters.

1.3 Outline of the Dissertation

The remainder of the dissertation focuses on solving the aforementioned research problems for smart and connected systems, which is organized as follows:

Chapter 2: Transfer Learning of Structures of Ordered Block Graphical Models Using Informative Priors

In this chapter, a transfer learning framework for learning interactions among stations in multistage connected manufacturing process is developed. The framework effectively takes advantages of heterogeneous data from different but related manufacturing processes to reveal the commonalities among these processes in a statistically rigorous way. The shared information is then put into a Bayesian presentation of mixture Gaussian models to complete the knowledge transfer towards a specific process to benefit the learning in complex connected manufacturing processes. The proposed method is general to interaction learning in multistage system for various characteristics and indexes. The

obtained interactions build the fundamental understanding of multistage connected systems and can be further applied in system modeling, inference and diagnostics.

Chapter 3: Approximate Key Performance Indicators Joint Distribution through Ordered Block Model and Pair Copula Construction

The key performance indicators (KPIs) are often used to characterize critical system objectives. The prediction and quantification of KPIs relationship will lead a better understanding of the system. In multistage manufacturing system, the KPIs are often random variables with complex interactions due to the intrinsically dynamic and random process in manufacturing system. In this chapter, a surrogate model based on pair-copula construction is proposed to capture the non-linear association among KPIs in manufacturing production lines through approximating the joint distribution of KPIs. The approximation employs physical knowledge of manufacturing production lines to remove redundant pair-copulas to reduce computational load. Theoretical properties are also derived to guarantee the iterative and analytical estimation of the joint distribution.

Chapter 4: Contamination Source Identification Based on Sequential Bayesian Approach for Water Distribution Network with Stochastic Demands

This chapter chooses the case study of contamination source identification in urban water distribution network to demonstrate the efficient algorithm in anomaly detection for smart and connected system. It is suggested to identify the contamination source in water distribution networks through real time sensor readings and efficient Bayesian strategies. The tremendous offline (prior) information of possible contamination source is organized into a concise tree structure to avoid combinatorial explosion. Then this information is used to compute the posterior probability of each source from the online sensor readings.

Chapter 5: Control of Key Performance Indicators of Manufacturing Production

Systems through Pair Copula Modeling and Stochastic Optimization

In this chapter, we would employ the KPI relationship quantified in Chapter 3 and formulate a KPI control framework. Our objective is to minimize the average operation cost in a system. The KPI control is framed into a stochastic optimization problem, where the noise distribution in the cost function depends on the decision variables. We propose to solve this stochastic optimization problem by employing the standard uniform distribution to link the variable relationship and the cost function so that the objective function is degenerated into an ordinary problem. The efficiency of the proposed method is investigated through various queuing systems.

Finally, Chapter 6 summarizes the contributions of the dissertation and discusses the future work.

2 TRANSFER LEARNING OF STRUCTURES OF ORDERED BLOCK GRAPHICAL MODELS USING INFORMATIVE PRIORS *

In this chapter, we use the ordered block model (OBM) to model the interactions in multistage manufacturing systems. The interactions in multistage manufacturing systems can be presented by the relationship among various random variables that characterize the system performance. The OBM is a special form of directed graphical model, which can characterize the random distribution in a compact form. In our approach, we focus on the learning of random variable inter-relationship in OBM, also known as structure learning, based on prior knowledge obtained from historical data. Since the historical manufacturing systems may not contain the same variables as those in the target system, the sample space of the graphical structure of the historical OBMs and the new OBM may be inconsistent. We deal with this issue by adding pseudo variables with probability normalization, then removing extra variables through marginalization to align the sample space between historical OBMs and the new OBM. The performance of the proposed method is illustrated and compared to conventional methods through numerical studies and a real car assembly process. The results show the proposed informative structure prior can effectively boost the performance of the structure learning procedure, especially when the data from the new OBM is small.

2.1 Introduction

Graphical model is a tool for exploiting structures in complex joint probability distributions. A graphical model includes a set of nodes V and a set of (directed or undirected) edges E.

*This chapter is based on the paper: **Wang C.**, Zhu X., Zhou S. and Zhou Y. "Learning of Structures of Ordered Block Graphical Models Using Informative Prior", submitted to *The Annals of Applied Statistics*.

Each node represents a random variable and each edge linking two nodes indicates the conditional dependence between the corresponding random variables. A graphical model often can describe a complex joint distribution compactly and allows the distribution to be constructed and utilized effectively [4]. The ordered block model (OBM) [5] is a special form of directed graphical model, where the nodes of the model are split into ordered blocks and an edge can only point from a node in a preceding block to a node in a later block. Fig. 2.1 shows a typical OBM, where q nodes are ordered and partitioned into numbered blocks. The directed edge is an indicator variable that represents the interaction from the parent node to the child node conditioning on all other nodes in the graph. For example, the directed edge from node 1 to 2 in Fig. 2.1 means the node 1 interacts the node 2 conditioning on other nodes. The OBM can find broad applications in practices, especially for sequential processes and experiments. For example, Bernard and Hartemink investigated the transcriptional regulatory network by setting the gene expression data into blocks according to the sample instants, and the interactions among the genes are learned under the blocked model [6]. Huang et al. also employed the OBM to investigate the interactions among the number of phone calls in a call center at different time periods, where the number of phone calls within 10-min are modeled as blocked random variables [7]. Another important example of OBM is the manufacturing multistage assembly process, where the dimensional quality measures at each work station can be modeled as nodes in the same block [8]. As discussed in [9], it is highly desirable to identify the joint distribution of the quality measures in multistage manufacturing processes for process control and quality improvement purposes.

This work concerns the learning of the structure of an OBM from data. The structure learning of a graphical model is a very challenging problem and extensive literature exists [6, 10, 11, 12, 13, 14, 15, 16]. The available approaches can be roughly classified into

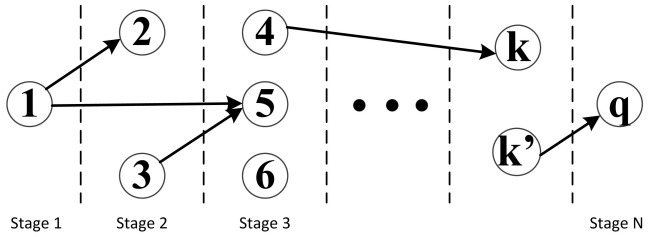


Figure 2.1: Diagram of an ordered block model.

methods based on conditional independence testing and methods based on scores [4]. The methods based on conditional independence testing are relatively easy to implement but suffer from the multiple testing issue, which makes the structure learning restrictive and inaccurate [17]. In this chapter, we focus on the score based method and use the Bayesian formulation of the learning problem as follows [18]:

$$g^* = \underset{g}{\operatorname{argmax}} P(\mathcal{G} = g|\mathbf{D}) \propto \underset{g}{\operatorname{argmax}} P(\mathbf{D}|\mathcal{G} = g) \cdot \pi(\mathcal{G} = g) \tag{2.1}$$

where $\mathbf{D}$ is the data collected from the nodes in the graph, $\mathcal{G}$ is a categorical random variable representing the unobservable structure of the graph, g is a specific realization of $\mathcal{G}$, $P(\mathbf{D}|\mathcal{G})$ is the data likelihood and $\pi(\mathcal{G})$ is the prior distribution of $\mathcal{G}$.

The concept of graph random variable $\mathcal{G}$ is very important and we would like to elaborate it further here. $\mathcal{G}$ is a finite categorical random variable and its sample space, defined as $\mathbf{\Omega}_{\mathcal{G}}$, contains all possible graph structures constructed by its nodes. Fig. 2.2 illustrates the sample space of $\mathcal{G}$ with 3 nodes 1, 2, and 4. Please note that here we only concern with OBM and thus there are only 4 possible realizations of $\mathcal{G}$. In the structure learning formulation (2.1), we want to compute the posterior distribution $P(\mathcal{G}|\mathbf{D})$ of $\mathcal{G}$, and then find the realization g^* that maximizes the posterior probability as the structure. The computation of $P(\mathcal{G}|\mathbf{D})$ will need data likelihood $P(\mathbf{D}|\mathcal{G})$ and the prior $\pi(\mathcal{G})$ of $\mathcal{G}$. In most existing literature, the prior $\pi(\mathcal{G})$ is simply selected as non-informative. For example,

$\pi(\mathcal{G}) = P(\mathcal{G} = g_i) = 0.25$, $i = 1, 2, 3, 4$ for $\mathcal{G}$ shown in Fig. 2.2.

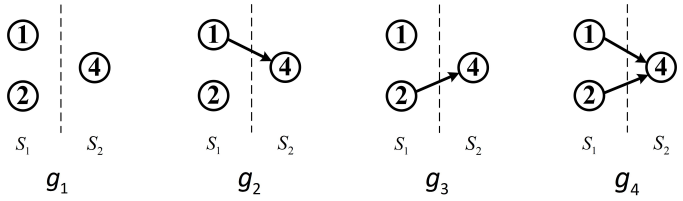


Figure 2.2: Illustration of graph random variable $\mathcal{G}$ with nodes 1, 2, 4 in two stages.

The non-informative prior makes the structure learning purely depend on the likelihood part, which often requires a large amount of data to achieve certain level of accuracy. Unfortunately, such a large dataset often is not available in practice. For example, in biomedical applications, the data collection can be very expensive and time-consuming [19]. In manufacturing applications, we often want to have the graphical model for quality and process control purposes. The nodes in the graph server as random variables representing the deviation from nominal values in the manufacturing processes. However, it is obvious that for a new process or a new product, the nodes data will be scarce. To deal with this situation, an alternative way is to employ knowledge learned from the data collected from existing similar processes to help the structure learning of the new process with limited data. Indeed, in manufacturing applications, it is rare that one manufacturing system only produces one type of product [20]. Instead, a product family, e.g., different car bodies, is often produced in the same manufacturing system. It is reasonable to expect that the graphical model structures for the production processes of different products in the same family will share some similarities due to the similarities in product features and production steps. This implies that to learn a new graphical model structure, we may take advantage of the historical data from other existing processes. The Bayesian framework in Equation (2.1) can naturally incorporate additional information into the structure learning: we can construct an informative prior $\pi(\mathcal{G})$ based on historical data to

benefit the structure learning process. This idea can be formulated as:

$$\begin{aligned} g^* &= \underset{g}{\operatorname{argmax}} P(\mathcal{G}|\mathbf{D}^1, \cdots, \mathbf{D}^n, \mathbf{D}) \\ &= \underset{g}{\operatorname{argmax}} P(\mathbf{D}|\mathcal{G}, \mathbf{D}^1, \cdots, \mathbf{D}^n) P(\mathcal{G}|\mathbf{D}^1, \cdots, \mathbf{D}^n) \\ &= \underset{g}{\operatorname{argmax}} P(\mathbf{D}|\mathcal{G}) P(\mathcal{G}|\mathbf{D}^1, \cdots, \mathbf{D}^n) \end{aligned} \tag{2.2}$$

where we denote the data collected from n historical processes as $\mathbf{D}^1, \cdots, \mathbf{D}^n$, and $P(\mathcal{G}|\mathbf{D}^1, \cdots, \mathbf{D}^n)$ is the informative prior we want to obtain from the historical data. Please note that from the second step to the third step in Equation (2.2), we used the fact that $\mathbf{D}$ is conditionally independent of $\mathbf{D}^1, \cdots, \mathbf{D}^n$ if the underlying graphical structure is given.

The basic idea shown in Equation (2.2) is straightforward. However, there are challenges in rigorously constructing the informative prior $P(\mathcal{G}|\mathbf{D}^1, \cdots, \mathbf{D}^n)$. One obvious challenge is that in general, the underlying sample spaces of $\mathbf{D}$ and $\mathbf{D}^i$, $i = 1, ..., n$ are different. For example, consider the process in Fig. 2.2, represented by $\mathcal{G}$, containing nodes 1, 2, and 4. If we have a historical process, represented by $\tilde{\mathcal{G}}^1$, contains nodes 1, 3, and 4 (1 and 3 in the first stage and 4 in the second stage). Then clearly although there is some similarity between $\mathcal{G}$ and $\tilde{\mathcal{G}}^1$ (i.e., both contain nodes 1 and 4 and their order is the same), $\mathbf{\Omega}_{\mathcal{G}} \neq \mathbf{\Omega}_{\tilde{\mathcal{G}}^1}$. Under rare situations, two processes will have exact the same set of nodes. How to transfer knowledge from similar yet different process is a challenging problem. In addition, we may have multiple historical processes and each of them has a different sample space. How to rigorously integrate the information in such a situation to obtain $P(\mathcal{G}|\mathbf{D}^1, \cdots, \mathbf{D}^n)$ is challenging. Very limited works exist studying the impacts of informative prior distributions on the graphical structure learning, and to the best of our knowledge no existing works address above mentioned challenges. Mansinghka et al

proposed a structure prior distribution construction method that can control the density of edges within the constructed graph [5]. Koski et al. further developed an algorithm that can set probabilities to control the existence of each edge in the prior of structures [21]. These two methods can quantitatively control the structure prior distribution. However, the structure prior is specified rather than learned from historical data in these methods. Flores et al. proposed to use expert knowledge elicitation for structure learning, which combines the domain knowledge from the experts to redistribute the prior structure distribution [22]. Again, their method cannot be integrated with learning algorithms based on observational data.

In this article, we propose a new structure learning method to effectively use the data collected from historical processes. As mentioned above, we treat $\mathcal{G}$ as a categorical random variable and thus the construction of $P(\mathcal{G}|\mathbf{D}^1, \cdots, \mathbf{D}^n)$ is to estimate the parameters of the categorical distribution using data $(\mathbf{D}^1, \cdots, \mathbf{D}^n)$. Denote $\mathcal{G}^i$ as the categorical random variable underlying the data $\mathbf{D}^i$, $i = 1, ..., n$. We first assume all $\mathcal{G}^i$, $i = 1, ..., n$, have the same sample space as $\mathcal{G}$ and let them follow the same categorical distribution $P(\mathcal{G})$. Because we only observe $(\mathbf{D}^1, \cdots, \mathbf{D}^n)$ and not the realization of $\mathcal{G}^i$, we formulate the problem into a problem of categorical distribution estimation with unobservable responses and provide a Bayesian estimation approach to solve it. In this way, we can obtain $P(\mathcal{G}|\mathbf{D}^1, \cdots, \mathbf{D}^n)$. In the general situation of inconsistent sample space among different processes, we first add "pseudo nodes" to make up the missing nodes between the historical process and the new process, and then marginalize different nodes between the historical process and the new process to get the consistent sample space. As a result, the data from the consistent sample space will be used in the estimation of $P(\mathcal{G})$. The numerical study and the real world example illustrate that the proposed learning approach can effectively help to identify the graphical structure with limited data. The contribution of our work is

to provide a rigorous Bayesian framework for incorporating data from related but different processes to benefit the structure learning of new process.

The rest of the chapter is organized as follows: Section 2.2 gives the problem formulation and presents the basic idea of the proposed graph structure learning method. Section 2.3 presents the detailed steps of constructing informative prior using historical data. The complete learning procedure is introduced in Section 2.4. Numerical studies will be conducted in Section 2.5 to show the effectiveness of the proposed method. Section 2.6 illustrates the application of the technique through a case study on the car body assembly process. Section 2.7 draws conclusion remarks.

2.2 Assumptions and outline of the basic approach

2.2.1 Notations and assumptions

We use upper case letter $\mathcal{G}$ to denote the graph random variable, and the lower case letter g to represent a specific realization of $\mathcal{G}$. We use the plain notation $\mathcal{G}$ and $\mathbf{D}$ to denote the graph random variable and the corresponding data for the new process under study, respectively. We use super-scripted symbol $\mathbf{D}^i, i \in \{1, 2, , \cdots, n\}$ to represent the data collected from n historical processes. To differentiate the sample space consistency between the historical process and the new process, we denote $\tilde{\mathcal{G}}^i$ as the graph random variable underlying $\mathbf{D}^i$ with arbitrary sample space, while $\mathcal{G}^i$ as the graph random variable transformed from $\tilde{\mathcal{G}}^i$ such that it has consistent sample space as the new process. Assume there are q nodes in $\mathcal{G}$. We let $\mathbf{Y} = (Y_1, Y_2, \cdots, Y_q)' \in \mathbf{R}^q$ be a random vector representing the corresponding random variables. For a specific node k in the graph, we denote $\mathcal{G}_k$ as a sub-graph that contains k as the child node and all nodes in stages before k. Please note in $\mathcal{G}_k$, only k can be the child node. In other words, edges in $\mathcal{G}_k$ can only be from

preceding nodes to k. Further, the sample space of $\mathcal{G}_k$ is defined as $\mathbf{\Omega}_{\mathcal{G}_k}$ containing all the possible realizations of $\mathcal{G}_k$, which are denoted as $g_{k,s}$, $s = 1, ..., |\mathbf{\Omega}_{\mathcal{G}_k}|$, where $|\cdot|$ is the cardinality of a set. We define an operator $\mathbf{V}(\cdot)$ on a graph, where $\mathbf{V}(\mathcal{G})$ is the set of all the nodes in $\mathcal{G}$. We use symbol "\" to represent excluding an element from a set. Thus, $\mathbf{V}(\mathcal{G}_k) \setminus k$ is the set of the nodes in stages before the node k in $\mathcal{G}_k$. We further define notations $\wp_k$ and $\bar{\wp}_k$ to represent the parent and non-parent nodes in the realization of $\mathcal{G}_k$. There is a directed edge from a parent node to k, while there is no edge between a non-parent node to k. We have $\wp_k \cup \bar{\wp}_k = \mathbf{V}(\mathcal{G}_k) \backslash k$. The number of edges that direct to a node is called the in-degree of a node.

With these notations, if we constrain the largest possible in-degree to node k be d in $\mathcal{G}_k$, then we have

$$|\mathbf{\Omega}_{\mathcal{G}_k}| = \sum_{i=0}^{\min(d, |\mathbf{V}(\mathcal{G}_k) \backslash k|)} \binom{|\mathbf{V}(\mathcal{G}_k) \setminus k|}{i} \tag{2.3}$$

The graphical structure learning is often built on assumptions with theoretical and data generation constraints [4]. We list the assumptions required for our work:

A1 We focus on Gaussian graphical model, i.e., $\mathbf{Y} \sim N_q(\mathbf{0}, \mathbf{\Sigma})$ with nonsingular covariance matrix $\mathbf{\Sigma}$.

A2 We assume the modularity property [23] for the graphical models, which means we have $P(\mathcal{G}) = \prod_{k=1}^{q} P(\mathcal{G}_k)$. This allows us to separate the global graph learning into q independent sub-graph learning.

A3 We assume the largest in-degree of the graphical models is d. This sparsity constraint [24] has been widely used in graphical model structure learning and can reduce the computational complexity.

The above assumptions are commonly used in graphical model learning literature and are not restrictive. This work concerns transferring knowledge from historical processes to the new process under study. We do not require the node set of $\mathcal{G}$ be identical to the historical processes, i.e., $\mathbf{V}(\mathcal{G})$ may not be equal to $\mathbf{V}(\tilde{\mathcal{G}}^i)$. We assume the overlapping nodes $\mathbf{V}(\mathcal{G}) \cap \mathbf{V}(\tilde{\mathcal{G}}^i)$ to be known and for the sake of notation convenience, we assume the nodes in $\mathbf{V}(\mathcal{G}) \cap \mathbf{V}(\tilde{\mathcal{G}}^i)$ have the same node index in $\mathcal{G}$ and $\tilde{\mathcal{G}}^i$, respectively. Also, we assume the nodes in $\mathbf{V}(\mathcal{G}) \cap \mathbf{V}(\tilde{\mathcal{G}}^i)$ follow consistent proceeding block order. In other words, if both nodes k and j are in $\mathbf{V}(\mathcal{G})$ and $\mathbf{V}(\tilde{\mathcal{G}}^i)$ and k is in the preceding stage of node j in $\mathcal{G}$, then k is also in the preceding stage of node j in $\tilde{\mathcal{G}}^i$. These assumptions essentially require there is some similarity between $\mathcal{G}$ and $\tilde{\mathcal{G}}^i$. After all, if there are no similarities, then there will be no information to be shared and transferred.

2.2.2 Outline of the basic approach for information sharing

Graph structure learning with informative prior is formulated in Equation (2.2), where we need to compute the informative prior and the data likelihood. The calculation of data likelihood $P(\mathbf{D}|\mathcal{G})$ follows ordinary procedures under the Bayesian linear model, which will be briefly reviewed in Section 2.4. Here we focus on the informative prior construction.

To compute the $P(\mathcal{G}|\mathbf{D}^1, \cdots, \mathbf{D}^n)$, we utilize the modularity assumption A2,

$$P(\mathcal{G}|\mathbf{D}^1, \cdots, \mathbf{D}^n) = \prod_{k=1}^{q} P(\mathcal{G}_k|\mathbf{D}^1, \cdots, \mathbf{D}^n) \tag{2.4}$$

Equation (2.4) divides the prior for $\mathcal{G}$ into prior for sub-structures $\mathcal{G}_k$ for $k = 1, ..., q$. If there is no historical processes containing node k, then we cannot get any information from historical data for the learning of $\mathcal{G}_k$. Thus, we can simply assign a non-informative

prior for $P(\mathcal{G}_k|\mathbf{D}^1, \cdots, \mathbf{D}^n)$ as

$$P(\mathcal{G}_k|\mathbf{D}^1, \cdots, \mathbf{D}^n) = \frac{1}{|\mathbf{\Omega}_{\mathcal{G}_k}|}, \text{ for } k \notin \bigcup_{i=1}^{n} \mathbf{V}(\tilde{\mathcal{G}}^i) \tag{2.5}$$

Please note that when we plug Equation (2.5) into Equation (2.4), this non-informative probability assignment will not change the ranking of probability of different realizations of $\mathcal{G}$ and thus the final learning result in Equation (2.2) will not be influenced. We shall focus on computing $P(\mathcal{G}_k|\mathbf{D}^1, \cdots, \mathbf{D}^n)$ when $k \in \bigcup_{i=1}^{n} \mathbf{V}(\tilde{\mathcal{G}}^i)$, where information sharing occurs. To facilitate easy exhibition, we first assume the node sets and hence the sample spaces of the historical process and the new process are identical (using $\mathcal{G}^i$ instead of $\tilde{\mathcal{G}}^i$). The case when they are not consistent will be addressed in details in Section 2.3.2.

The essential idea enabling information sharing between $\mathcal{G}$ and $\mathcal{G}^i$, $i = 1, ..., n$ is that *we assume $\mathcal{G}$ and $\mathcal{G}^i$ follow the same underlying categorical distribution* $\pi(\mathcal{G})$. We can utilize the data from $\mathcal{G}^i$, $i = 1, ..., n$, to estimate the parameters of the categorical distribution $\pi(\mathcal{G})$ and then the estimated distribution will in turn provides the informative prior for the learning of $\mathcal{G}$. In this way, the information in the data collected from $\mathcal{G}^i$, $i = 1, ..., n$, are shared in the structure learning of $\mathcal{G}$ and the common categorical distribution $\pi(\mathcal{G})$ is the vehicle carrying the shared information. This idea is further illustrated in Fig. 2.3, where we use a graph containing three nodes (1, 2 and 4) as an example. In Fig. 2.3, we want to estimate the event probabilities (p_1, p_2, p_3, p_4) of the common categorical distribution $\pi(\mathcal{G}_4)$ based on the observed data $\mathbf{D}^1, \mathbf{D}^2, ..., \mathbf{D}^n$ and then it will be used as the informative prior $P(\mathcal{G}_4|\mathbf{D}^1, \cdots, \mathbf{D}^n)$.

In Fig. 2.3, each $\mathcal{G}_4^i$ is a random variable following distribution $\pi(\mathcal{G}_4)$. The $\mathbf{D}^i$ is the data collected from the corresponding graphical model, which is an unobservable realization of $\mathcal{G}_4^i$. If the samples from $\pi(\mathcal{G}_4)$, i.e., the specific realizations of $\mathcal{G}_4^1$, $\mathcal{G}_4^2$,..., and $\mathcal{G}_4^n$, are directly observable, then the estimation of (p_1, p_2, p_3, p_4) will be straightforward. Unfortunately,

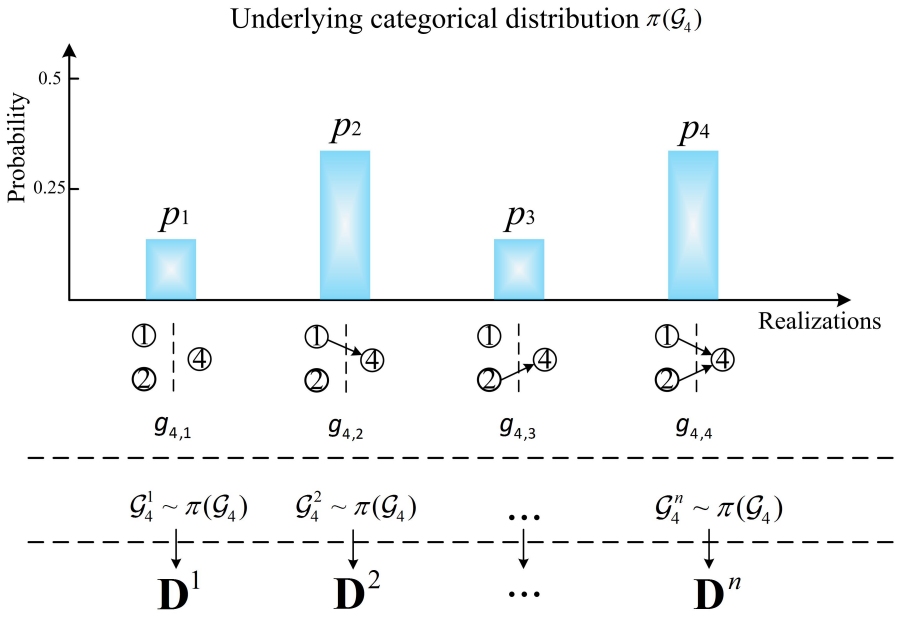


Figure 2.3: The data generation framework.

the challenge is that such samples are not observable. Rather, we only observe $\mathbf{D}^1, \cdots, \mathbf{D}^n$, whose distribution is related with such samples. This problem can be framed into a general problem of estimating categorical distributions with indirect observations. Maximum likelihood estimation (MLE) method could be applied by transforming the above problem into a mixture Gaussian model estimation problem [25]. However, the objective function of MLE is non-convex, and it often suffers singularity and identifiability issues. Alternatively, we propose to use a Bayesian approach to estimate the categorical distribution. The Bayesian approach treats the event probabilities (e.g., p_1, p_2, p_3, p_4 in Fig. 2.3) as random variables rather than fixed values, and calculates the posterior of these random variables based on data $\mathbf{D}^1, \mathbf{D}^2, ..., \mathbf{D}^n$ and a non-informative prior. To compute the posterior probability and due to the indirect observations, we need to consider all the possible combinations of the realizations of $\mathcal{G}_4^i$, $i = 1, ..., n$. It is easy to see that the number of such combinations is $|\mathbf{\Omega}_{\mathcal{G}_k}|^n$. For example, for $\mathcal{G}_4$ in Fig. 2.3, the combination of realization is

Table 2.1: Possible realizations of $(\mathcal{G}_4^1, \mathcal{G}_4^2)$.

Index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$\mathcal{G}_4^1$	$g_{4,1}$	$g_{4,1}$	$g_{4,1}$	$g_{4,1}$	$g_{4,2}$	$g_{4,2}$	$g_{4,2}$	$g_{4,2}$	$g_{4,3}$	$g_{4,3}$	$g_{4,3}$	$g_{4,3}$	$g_{4,4}$	$g_{4,4}$	$g_{4,4}$	$g_{4,4}$
$\mathcal{G}_4^2$	$g_{4,1}$	$g_{4,2}$	$g_{4,3}$	$g_{4,4}$	$g_{4,1}$	$g_{4,2}$	$g_{4,3}$	$g_{4,4}$	$g_{4,1}$	$g_{4,2}$	$g_{4,3}$	$g_{4,4}$	$g_{4,1}$	$g_{4,2}$	$g_{4,3}$	$g_{4,4}$

4^n. More specifically, assume $n = 2$. Then we have 16 possible realizations of $(\mathcal{G}_4^1, \mathcal{G}_4^2)$ as shown in Table 2.1 with the first row as the index of realizations.

This illustrates that the collected data $(\mathbf{D}^1, \mathbf{D}^2)$ may come from any of the 16 combinations of graph realizations.

For the sake of clear logic flow, we put the description of the Bayesian approach into appendix 2.8.1 and only present the result as below:

$$\begin{aligned} &P(\mathcal{G}_k = g_{k,s^*} | \mathbf{D}^1, \cdots, \mathbf{D}^n) \\ &\propto \sum_{j=1}^{|\mathbf{\Omega}_{\mathcal{G}_k}|^n} \left\{ \frac{\Gamma(|\mathbf{\Omega}_{\mathcal{G}_k}|)}{\Gamma(\sum_{l=1}^{|\mathbf{\Omega}_{\mathcal{G}_k}|} c_l + |\mathbf{\Omega}_{\mathcal{G}_k}|)} \cdot \prod_{l=1}^{|\mathbf{\Omega}_{\mathcal{G}_k}|} \Big(\Gamma(1 + c_l)\Big) \cdot \prod_{i=1}^{n} P(\mathcal{G}_k^i = g_{k,S(i,j)} | \mathbf{D}^i) \right\} \end{aligned} \tag{2.6}$$

where $g_{k,S(i,j)}$ is the realization of $\mathcal{G}_k^i$ in the jth realization combination and $S(i,j)$ is the corresponding realization index. Using the realization combinations shown in Table 1 as an example. S is actually a 2×16 matrix consisting of the second subscript index of g in Table 1. For instance, we have $g_{4,S(2,10)} = g_{4,2}$ and thus $S(2,10)$ is the index 2. With this notation, we further have $c_l = \mathbf{1}(s^* = l) + \sum_{i=1}^{n} \mathbf{1}(S(i,j) = l)$, where $\mathbf{1}(\cdot)$ is the indicator function; $\Gamma(\cdot)$ is the Gamma function. Please note that in Equation (2.6), we need to sum over $|\mathbf{\Omega}_{\mathcal{G}_k}|^n$ possible realization combinations, which could be a large value. This complication is resulted from the fact that the realization of $\pi(\mathcal{G}_k)$ is not directly observable and we need to try all the possible values of the sample in the Bayesian approach. For large number of n, numerical approximations to Equation (2.6) is available [26]. We also discuss the computational issue in the last section of the chapter.

2.3 Construct informative prior distribution

Equation (2.6) provides the fundamental methods for constructing the informative prior from multiple historical processes. Combining with Equation (2.2), we can fulfill the transfer learning of graphical model structures. In this section, we present a method of computing $P(\mathcal{G}_k^i = g_{k,s}|\mathbf{D}^i)$ in (2.6), where s is an arbitrary realization index of $\mathcal{G}_k^i$. We will also discuss how to address the issue of sample space inconsistency.

2.3.1 Determine $P(\mathcal{G}_k^i = g_{k,s}|\mathbf{D}^i)$

Determining $P(\mathcal{G}_k^i = g_{k,s}|\mathbf{D}^i)$ is a general problem in structure learning of graphical models. Several methods using the Bayes rule are available [27]. However, most of these methods require the specification of the prior distribution of the parameters of the graphical model and the performance is sensitive to such specification [4, 27, 28]. In this work, we propose to degrade the problem from graph level to edge level and use regression based hypothesis tests to specify $P(\mathcal{G}_k^i = g_{k,s}|\mathbf{D}^i)$.

Specifically, we can express the $P(\mathcal{G}_k^i = g_{k,s}|\mathbf{D}^i)$ as follows:

$$P(\mathcal{G}_k^i = g_{k,s}|\mathbf{D}^i) = P(\mathbf{E}_{k,s} = \mathbf{1}, \bar{\mathbf{E}}_{k,s} = \mathbf{0}|\mathbf{D}^i) \tag{2.7}$$

where $\mathbf{E}_{k,s} = [E^1_{\wp_{k,s}}, \cdots, E^l_{\wp_{k,s}}]$ with $l = |\wp_{k,s}|$; $\bar{\mathbf{E}}_{k,s} = [E^1_{\bar{\wp}_{k,s}}, \cdots, E^r_{\bar{\wp}_{k,s}}]$ with $r = |\bar{\wp}_{k,s}|$; $\wp_{k,s}$ and $\bar{\wp}_{k,s}$ are the parent set and non-parent set of $g_{k,s}$, respectively; $E^w_{\wp_{k,s}}$ is the binary edge indicator variable corresponding to the edge from the wth node in $\wp_{k,s}$ to the child and similarly, $E^u_{\bar{\wp}_{k,s}}$ corresponds to the edge from the uth node in $\bar{\wp}_{k,s}$ to the child. With this notation, Equation (2.7) simply states that edges exist from parent nodes and do not exist from non-parent nodes to the child node (i.e., node k) in $g_{k,s}$. We can further decompose the probability in Equation (2.7) using the following lemma.

Lemma 1: For the sub-graph $g_{k,s}$, we can decompose $P(\mathcal{G}_k^i = g_{k,s}|\mathbf{D}^i)$ as $l + r$ multiplications as follows:

$$\begin{aligned} & P(\mathcal{G}_k^i = g_{k,s}|\mathbf{D}^i) \\ & \propto \prod_{w=1}^{l} P(E^w_{\wp_{k,s}} = 1|\bar{\mathbf{E}}_{k,s} = \mathbf{0}, \mathbf{D}^i) \cdot \\ & \quad \prod_{u=1}^{r} \left(1 - P(E^u_{\bar{\wp}_{k,s}} = 1|\bar{\mathbf{E}}^{-u}_{k,s} = \mathbf{0}, \mathbf{D}^i)\right) \end{aligned} \tag{2.8}$$

where $\bar{\mathbf{E}}^{-u}_{k,s}$ is the same indicator vector as $\bar{\mathbf{E}}_{k,s}$ except for the absence of the edge indicator variable $E^u_{\bar{\wp}_{k,s}}$.

The proof of lemma 1 is presented in appendix 2.8.2. Each $P(\cdot|\cdot)$ in Equation (2.8) represents the probability of the presence of a specific single edge given that a subset of edges do not exist. To test the presence of an edge, we can establish a conditional normal linear regression model and test if a corresponding coefficient is zero [29]. This property motivates us to use the p-value of the hypothesis testing on the coefficient to represent our belief on the probability of the existence of that edge in the graph and thus quantify each $P(\cdot|\cdot)$ in Equation (2.8). We use an example to illustrate how to set up the corresponding regression model and the hypothesis test.

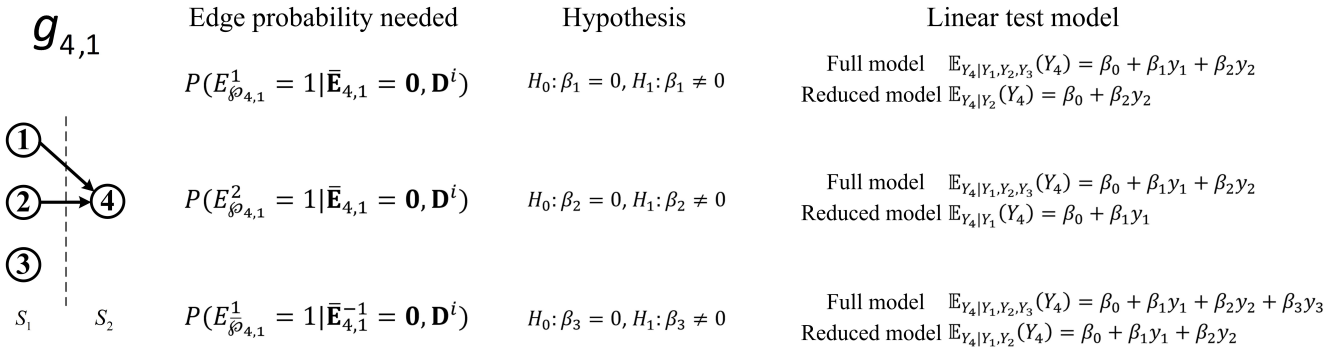


Figure 2.4: Regression based hypothesis testing for edge presences.

Figure 2.4 shows a sub-graph realization $g_{4,1}$ containing 4 nodes (1,2,3,4) and the node 4 is the child node. The parent and non-parent set for $g_{4,1}$ is $\wp_{4,1} = \{1, 2\}$ and $\bar{\wp}_{4,1} = \{3\}$, respectively. According to the lemma 1, we need three $P(\cdot|\cdot)$ to obtain the $P(\mathcal{G}_4 = g_{4,1}|\mathbf{D}^i)$, which are shown as the 'Edge probability needed' column in Fig. 2.4. The corresponding hypothesis testing and the associated full and reduced models are shown in the third and the fourth columns of Fig. 2.4. From this example, we can see it is straightforward to test the presence of an edge: we just remove the nodes that do not have an edge to child (as indicated by the conditioning part of $P(\cdot|\cdot)$) to construct the full model and then remove the node corresponding to the edge we want to test from the full model to construct the reduced model. Note that the β_0 in the test could be set to 0 due to the assumption A1. The p-value of a hypothesis testing is a random variable that quantifies the power of rejecting the null hypothesis. Intuitively, the higher the p-value from the testing, the weaker the evidence to reject the null hypothesis, which results in the lower possibility of the edge presence. Based on this idea, we can characterize the probability of the presence of an edge through p-value of the corresponding hypothesis testing as:

$$\begin{aligned} &P(E_j = 1|Q_j = q_j) \\ &= \frac{f(Q_j = q_j|E_j = 1)P(E_j = 1)}{f(Q_j = q_j)} \\ &= \frac{f(Q_j = q_j|E_j = 1)P(E_j = 1)}{f(Q_j = q_j|E_j = 1))P(E_j = 1) + f(Q_j = q_j|E_j = 0)P(E_j = 0)} \end{aligned} \tag{2.9}$$

where E_j is the edge indicator variable for the edge from node j to the child, Q_j is the random variable representing the corresponding p-value of the hypothesis testing, $f(\cdot)$ is the probability density function, and $P(E_j)$ is the prior probability mass function of E_j. To compute $P(E_j = 1|Q_j = q_j)$, we need the prior distribution $P(E_j)$, and $f(Q_j = q_j|E_j = 1)$, $f(Q_j = q_j|E_j = 0)$, which are the p-value distribution given the alternative and the null,

respectively. It is known that given the null hypothesis is true, the p-value is uniformly distributed on [0,1] [30]. Thus, we can simply assign

$$f(Q_j = q_j | E_j = 0) = 1, \quad q_j \in [0,\ 1] \tag{2.10}$$

The p-value distribution under alternative is complicated and there is no explicit result. We only know that given the alternative is true, the p-value should tend to be near 0 [30]. Here we adopt the truncated exponential distribution as the p-value distribution under alternative case:

$$f(Q_j = q_j | E_j = 1) = \lambda_j \cdot \frac{e^{-\lambda_j \cdot q_j}}{1 - e^{-\lambda_j}} \tag{2.11}$$

where λ_j is the scaling parameter of the truncated exponential distribution. This density function is a monotonically decreasing function on [0, 1]. Although this distribution is generally not the true distribution of the p-value, it is shown that it can quantify the link between the p-value and our belief on the probability of edge presence and thus serves the purpose of graph structure learning well [6].

The specification of the prior distribution $P(E_j = 1)$ is relatively straightforward. Let $P(E_j = 1) = \alpha_j$. Then if we have specific knowledge on the presence of certain edges, we can assign the corresponding α_j, otherwise we can choose the non-informative prior as $\alpha_j = 0.5$.

With above specifications, Equation (2.9) becomes

$$f(E_j = 1 | Q_j = q_j) = \frac{\lambda_j \cdot \alpha_j \cdot e^{-\lambda_j \cdot q_j}}{\lambda_j \cdot \alpha_j \cdot e^{-\lambda_j \cdot q_j} + (1 - e^{-\lambda_j}) \cdot (1 - \alpha_j)} \tag{2.12}$$

To compute Equation (2.12), we also need the value of the parameter λ_j. From Equation (2.11), we can see that a small value of λ_j close to zero leads to a "flat" p-value distribution, while a large value of λ_j leads to a sharp distribution of p-value. In practices,

it is sometimes hard to pick a single value for λ_j. Here we use a robust approach that avoids the selection of a single value and instead marginalizes over λ_j. We assume the λ_j is uniformly distributed over the interval $[\lambda_j^L, \lambda_j^H]$ and integrate λ_j out of Equation (2.12):

$$\begin{aligned} & f(E_j = 1 | Q_j = q_j) \\ & = \frac{1}{\lambda_j^H - \lambda_j^L} \int_{\lambda_j^L}^{\lambda_j^H} \frac{\lambda_j \cdot \alpha_j \cdot e^{-\lambda_j \cdot q_j}}{\lambda_j \cdot \alpha_j \cdot e^{-\lambda_j \cdot q_j} + (1 - e^{-\lambda_j}) \cdot (1 - \alpha_j)} \mathrm{d}\lambda_j \end{aligned} \tag{2.13}$$

The λ_j^L can be selected as near 0 to represent the possible "flat" p-value distribution, and the value of λ_j^H depends on our believe on the power of the test. If we have a large dataset and we believe the data quality is good, we can select a relatively large value of λ_j^H. The influence on the structure learning performance by choosing λ_j^L and λ_j^H will be discussed in Section 2.5.

Finally, each edge probability in Equation (2.8) can be calculated by Equation (2.13) through the p-values obtained by the hypothesis test procedures in Fig. 2.4. We can thus specify the term $P(\mathcal{G}_k^i = g_{k,s} | \mathbf{D}^i)$.

2.3.2 Construct consistent sample space

The sample space of the graph random variable for historical processes is assumed to be identical to that of $\mathcal{G}_k$ in previous discussion. However, as we introduced in Section 2.1, in most cases, the node set and hence the sample space of $\mathcal{G}_k$ and historical graphical random variables will be different. Recall the notation $\tilde{\mathcal{G}}_k^i$ in Section that represents the general historical graphical random variable with arbitrary sample space. Here we propose a method of adding "pseudo nodes" with probability normalization and removing extra nodes through marginalization to align the sample space between $\tilde{\mathcal{G}}_k^i$ and $\mathcal{G}_k$. The basic idea is illustrated in Fig. 2.5.

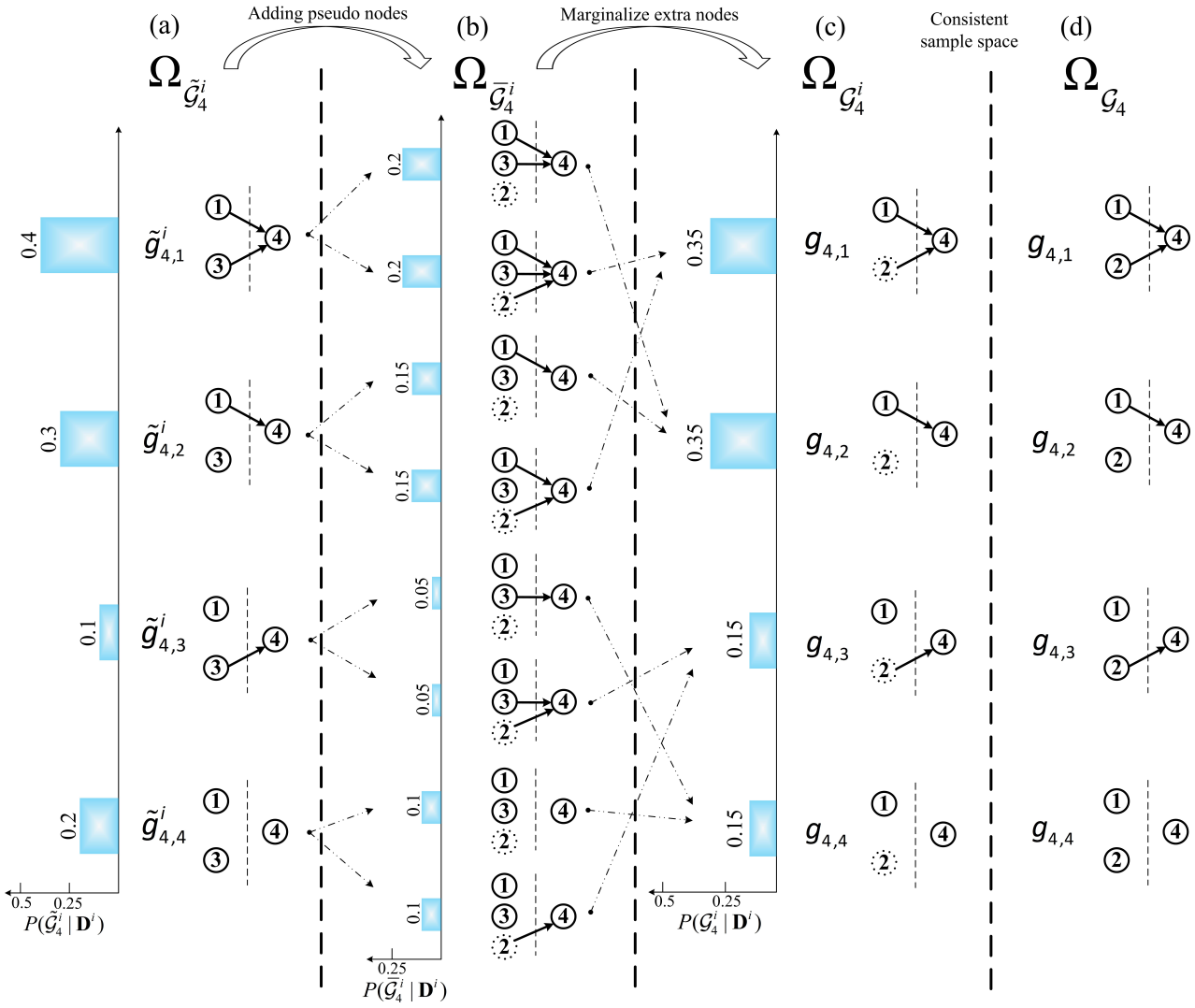


Figure 2.5: Structure prior learning from the ith historical process to the new process.

In Fig. 2.5, $\mathbf{\Omega}_{\tilde{\mathcal{G}}_4^i}$ shown in (a) is obviously different from $\mathbf{\Omega}_{\mathcal{G}_4}$ shown in (d). Compared with $\mathcal{G}_4$, $\tilde{\mathcal{G}}_4^i$ does not have node 2 but has an extra node 3. To construct a consistent sample space from $\mathbf{\Omega}_{\tilde{\mathcal{G}}_4^i}$ to $\mathbf{\Omega}_{\mathcal{G}_4}$, we first add a pseudo node 2 into $\tilde{\mathcal{G}}_4^i$ and expand its sample space to $\bar{\mathcal{G}}_4$ as shown by the step (a)→(b) in Fig. 2.5. Because we do not have any prior knowledge about the relationship between the pseudo node with other existing nodes in $\mathcal{G}_4^i$, the probabilities on $\bar{\mathcal{G}}_4$ simply duplicates that for $\mathbf{\Omega}_{\tilde{\mathcal{G}}_4^i}$ with a normalization as shown in (b). In the second step, we marginalize the extra node, which is node 3, in $\bar{\mathcal{G}}_4^i$. This step is shown as (b)→(c) in Fig. 2.5. With the marginalization step, the sample space in (c), denoted as $\mathbf{\Omega}_{\mathcal{G}_4^i}$, will be the same as $\mathbf{\Omega}_{\mathcal{G}_4}$ and the probability $P(\mathcal{G}_4^i|\mathbf{D}^i)$ can then be used in Equation (2.6).

Below we give a general result for dealing with the inconsistency between the probability sample space $\mathbf{\Omega}_{\tilde{\mathcal{G}}_k^i}$ and $\mathbf{\Omega}_{\mathcal{G}_k^i}$, where $\mathbf{\Omega}_{\mathcal{G}_k^i} = \mathbf{\Omega}_{\mathcal{G}_k}$:

$$P(\mathcal{G}_k^i = g_{k,s}|\mathbf{D}^i) = \frac{\sum_{j\in s^+} P(\tilde{\mathcal{G}}_k^i = \tilde{g}_{k,j}^i|\mathbf{D}^i)}{2^{a^*}} \tag{2.14}$$

where $a^* = \left|\mathbf{V}(\mathcal{G}_k^i)\backslash\Big(\mathbf{V}(\mathcal{G}_k^i)\bigcap\mathbf{V}(\tilde{\mathcal{G}}_k^i)\Big)\right|$ represents the number of pseudo nodes we need to add to $\tilde{\mathcal{G}}_k$, s^+ contains the indices of a selected set of realizations of $\tilde{\mathcal{G}}_k^i$ such that $g_{k,s} \subseteq \bar{g}_{k,j}, j \in s^+$. Clearly the operations in the numerator achieves the marginalization step and dividing by 2^{a^*} is the normalization step. Taking the example in Fig. 2.5, we can specify $P(\mathcal{G}_4^i = g_{4,1}|\mathbf{D}^i)$ from Equation (2.14) as: $P(\mathcal{G}_4^i = g_{4,1}|\mathbf{D}^i) = \frac{P(\tilde{\mathcal{G}}_4^i=\tilde{g}_{4,1}^i|\mathbf{D}^i)+P(\tilde{\mathcal{G}}_4^i=\tilde{g}_{4,2}^i|\mathbf{D}^i)}{2} = \frac{0.4+0.3}{2} = 0.35$.

With Equation (2.14) we can transform the probability space of an arbitrary graph random variable into a probability space that is consistent with the current graphical model under study and then use it for information sharing.

2.4 The overall learning procedure

Section 2.3 introduces the way to obtain the informative prior for the graph structure learning. To complete the learning process, we still need to have the data likelihood part. Due to the assumption A1, this likelihood part can be analytically obtained under the Bayesian linear model with conjugate priors [31]. We will briefly introduce the idea of constructing the Bayesian linear model and directly give the likelihood result. Some derivation details are in the appendix 2.8.3.

According to the assumption A1, for a sub-graph realization g_{k,s^*}, we can have the following relationships among nodes:

$$\begin{aligned} Y_k &= \mathbf{Z}_{\wp_{k,s^*}} \boldsymbol{\beta}_{k,s^*} + \varepsilon_{k,s^*} \\ \varepsilon_{k,s^*} &\sim N(0, \sigma^2_{k,s^*}) \end{aligned} \tag{2.15}$$

where $\boldsymbol{\beta}_{k,s^*}$ is the coefficient vector for the parent nodes in g_{k,s^*}, and σ^2_{k,s^*} is the variance for Y_k. Note that we use the vector $\mathbf{Z}_{\wp_{k,s^*}}$ to represent a sample of parent nodes in g_{k,s^*}. Equation (2.15) describes the conditional distribution of Y_k given the $\mathbf{Z}_{\wp_{k,s^*}}$. The basic idea of Bayesian linear model is to assign a Normal-inverse-Gamma distribution for the parameters $\boldsymbol{\beta}_{k,s^*}$ and σ^2_{k,s^*}. As a result, the $P(\boldsymbol{\beta}_{k,s^*}, \sigma^2_{k,s^*})$ will be the conjugate prior for $P(Y_k|\boldsymbol{\beta}_{k,s^*}, \sigma^2_{k,s^*})$, which makes the likelihood $P(\mathbf{D}|\mathcal{G})$ analytically available.

We denote $\boldsymbol{\mu}_{\beta_{k,s^*}}$ and $\mathbf{V}_{\beta_{k,s^*}}$ as the mean vector and covariance matrix in the normal distribution for $\boldsymbol{\beta}_{k,s^*}$, the γ_{k,s^*} and η_{k,s^*} as the parameters in the inverse Gamma distribution $IG(.,.)$ for σ^2_{k,s^*}. Then, we can have likelihood term in terms of $\boldsymbol{\mu}_{\beta_{k,s^*}}, \mathbf{V}_{\beta_{k,s^*}}, \gamma_{k,s^*}, \eta_{k,s^*}$

and $\mathbf{D}$ as:

$$P(\mathbf{D}|\mathcal{G}) = \prod_{k=1}^{q} \frac{\eta_{k,s^*}^{\gamma_{k,s^*}} \Gamma(\gamma_{k,s^*} + \frac{v}{2}) \sqrt{|\mathbf{V}^*_{k,s^*}|}}{(2\pi)^{\frac{v}{2}} \Gamma(\gamma_{k,s^*}) \sqrt{|\mathbf{V}_{\beta_{k,s^*}}|}} \Big[\eta_{k,s^*} + \frac{1}{2} \{ \mathbf{Y}_k^T \mathbf{Y}_k + \boldsymbol{\mu}^T_{\beta_{k,s^*}} \mathbf{V}^{-1}_{\beta_{k,s^*}} \boldsymbol{\mu}_{\beta_{k,s^*}} - {\boldsymbol{\mu}^*_{k,s^*}}^T {\mathbf{V}^*_{k,s^*}}^{-1} \boldsymbol{\mu}^*_{k,s^*} \} \Big]^{-(\gamma_{k,s^*} + \frac{v}{2})} \tag{2.16}$$

where

$$\begin{aligned} \boldsymbol{\mu}^*_{k,s^*} &= (\mathbf{V}^{-1}_{\beta_{k,s^*}} + \tilde{\mathbf{Z}}^T_{\wp_{k,s^*}} \tilde{\mathbf{Z}}_{\wp_{k,s^*}})^{-1} (\mathbf{V}^{-1}_{\beta_{k,s^*}} \boldsymbol{\mu}_{\beta_{k,s^*}} + \tilde{\mathbf{Z}}^T_{\wp_{k,s^*}} \mathbf{Y}_k) \\ \mathbf{V}^*_{k,s^*} &= (\mathbf{V}^{-1}_{\beta_{k,s^*}} + \tilde{\mathbf{Z}}^T_{\wp_{k,s^*}} \tilde{\mathbf{Z}}_{\wp_{k,s^*}})^{-1} \end{aligned} \tag{2.17}$$

where $\tilde{\mathbf{Z}}_{\wp_{k,s^*}} = \{\mathbf{Z}^1_{\wp_{k,s^*}}, \cdots, \mathbf{Z}^v_{\wp_{k,s^*}}\}^T$ represents the v samples of the random variable vector $\mathbf{Z}_{\wp_{k,s^*}}$. Once we obtain the informative prior and the data likelihood, we can conduct the structure learning according to Equation (2.2).

To benefit the new OBM structure learning, the prior must be correctly specified. A mis-specified prior will mislead the new OBM learning process. However, in the real OBM structure learning problem, it is not guaranteed the prior knowledge about OBMs is correctly specified. Under some extreme situations, the learning process with wrong prior knowledge may need more data and time to find the underlying OBM comparing with the learning with non-informative prior. To overcome the non-conformity between the prior knowledge and the new OBM, we propose a prior correcting algorithm to correct the potential misleading information in the informative prior.

The correcting algorithm is based on the Bayes factor, which is a widely used criterion for model selection in Bayesian analysis [32]. The general idea of Bayes factor is to compare the ratio of data likelihood of the two models: $b_{1,2} = \frac{P(\mathbf{D}|M_1)}{P(\mathbf{D}|M_2)}$, where $P(\mathbf{D}|M_m)$ is the data likelihood for model $M_m, m = 1, 2$. If $b_{1,2} > 1$, we choose model 1, otherwise we choose model 2. In the OBM structure learning, we can first use the proposed method and

the method with non-informative prior to learn the graph structure. The two methods give two learned graph structures denoted as g_{I} and g_{II}, respectively. Then, we can use the Bayes factor to choose between the two learning results. With this procedure, when the prior knowledge is misleading, the final learning result will degenerate to the non-informative method. In this chapter, we directly use the likelihood in Equation (2.16) as the non-informative method.

The informative prior based structure learning methodology is summarized as follows:

Algorithm 1 Graphical structure learning based on informative prior

---

Main algorithm:

1. In the new OBM with q nodes, employ expert knowledge and engineering experience, determine the maximum indgree d. Identify the n related historical graphs that have enough data for structure prior learning;

2. Obtain the sub-graph informative prior $P(\mathcal{G}_1|\mathbf{D}^1,\cdots,\mathbf{D}^n)$, $\cdots$,$P(\mathcal{G}_q|\mathbf{D}^1,\cdots,\mathbf{D}^n)$ for every node in the new graph through the sub algorithm, and multiply them together to get $P(\mathcal{G}|\mathbf{D}^1,\cdots,\mathbf{D}^n)$;

3. Split the $\mathbf{D}$ into $\mathbf{D}^{0.8}$ and $\mathbf{D}^{0.2}$ that represent the 80% and 20% of the new process data, respectively;

4. Construct the Bayesian score with $P(\mathbf{D}^{0.8}|\mathcal{G})$ and $P(\mathcal{G}|\mathbf{D}^1,\cdots,\mathbf{D}^n)$, and find the underlying OBM according to Equation (2.2). Denote the learned OBM as g_{I};

5. Construct the Bayesian score with $P(\mathbf{D}^{0.8}|\mathcal{G})$ and an uniformly distributed prior $P_{uni}(\mathcal{G})$, and find the underlying OBM according to Equation (2.2). Denote the learned OBM as g_{II};

6. Use the Bayes factor $b_{g_{\mathrm{I}},g_{\mathrm{II}}} = \frac{P(\mathbf{D}^{0.2}|g_{\mathrm{I}})}{P(\mathbf{D}^{0.2}|g_{\mathrm{II}})}$ to select the final learn graph $g_{OBM} = g_{\mathrm{I}} \cdot \mathbf{1}(b_{g_{\mathrm{I}},g_{\mathrm{II}}} > 1) + g_{\mathrm{II}} \cdot \mathbf{1}(b_{g_{\mathrm{I}},g_{\mathrm{II}}} \leq 1)$.

Output: g_{OBM}.

Sub algorithm for correcting informative prior:

If $k \notin \bigcup_{i=1}^{n} \mathbf{V}(\tilde{\mathcal{G}}^i)$

Use Equation (2.5) to get the $P(\mathcal{G}_k|\mathbf{D}^1, \cdots, \mathbf{D}^n)$.

Else if sample space consistent

1. Use Equation (2.8) to calculate $P(\mathcal{G}_k^i = g_{k,s}|\mathbf{D}^i)$ for $i = 1, \cdots, n$;
2. Plug all the $P(\mathcal{G}_k^i = g_{k,s}|\mathbf{D}^i)$ into Equation (2.6) to get $P(\mathcal{G}_k|\mathbf{D}^1, \cdots, \mathbf{D}^n)$.

Else

1. Use Equation (2.8) to calculate $P(\tilde{\mathcal{G}}_k^i = \tilde{g}_{k,s}^i|\mathbf{D}^i)$ for $i = 1, \cdots, n$;
2. Use Equation (2.14) to get $P(\mathcal{G}_k^i = g_{k,s}|\mathbf{D}^i)$;
3. Plug all the $P(\mathcal{G}_k^i = g_{k,s}|\mathbf{D}^i)$ into Equation (2.6) to get $P(\mathcal{G}_k|\mathbf{D}^1, \cdots, \mathbf{D}^n)$.

End

2.5 Numerical studies

In this section, we apply the proposed method in learning graph structures and compare the performance with the widely used non-informative prior metric [15]. The simulation procedure is as follows: First, we choose a new process layout and n historical process layouts, and evaluate the sample space consistency of each sub-graph between the new process and historical processes. Then, the sub-algorithm introduced in the algorithm 1 is employed to obtain the sub-graph informative prior $P(\mathcal{G}_1|\mathbf{D}^1, \cdots, \mathbf{D}^n)$, $\cdots$,$P(\mathcal{G}_q|\mathbf{D}^1, \cdots, \mathbf{D}^n)$. The learning of the new graph is completed by using the main algorithm in the algorithm 1. To test the robustness of the proposed method and show the effectiveness of the prior correction method, the same informative prior will be used in learning different new processes. Some of the new process structures may not be presented by the informative prior, which simulates the non-conformity between the prior and the graph structure. To obtain different new processes, we randomize the edge existence in the new process with

fixed nodes layout. Finally, the performance of the structure learning using the same informative prior will be evaluated with different new processes. In our simulations, we choose two historical graphs ($n = 2$) as shown in Fig. 2.6, whose nodes layout is inspired by the real case study in the following section. Nodes in the graph will follow Equation (2.15) to construct a multivariate normal distribution, where we set all $\sigma = 20$ and $\boldsymbol{\beta} = 1$ for the directed edges. We set the edge existence probability $\alpha = 0.5$ for all edges. In the practical applications, the α can be adjusted based on expert knowledge to improve the efficiency of the learning process. The sparsity constraint is set as $d = 3$. The new graph layout and an example of the new graph structure is shown in Fig. 2.7.

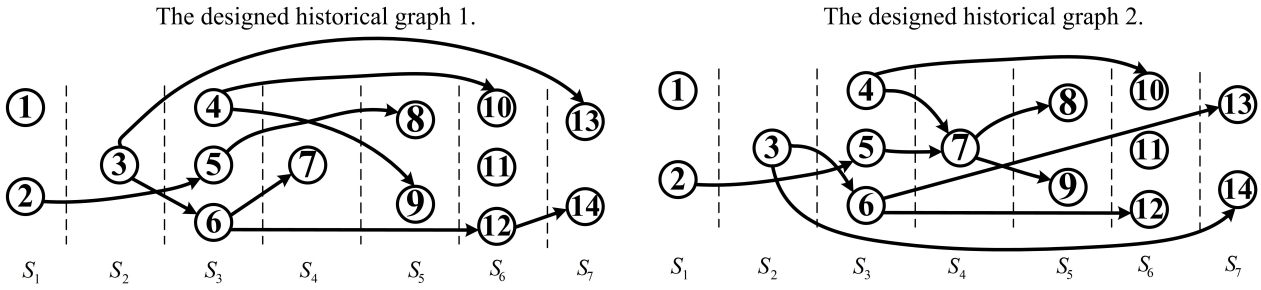


Figure 2.6: The designed historical graphs.

We follow the steps in Section 2.4 to construct the informative prior and learn the new graph structure. Here we introduce an index to describe the power of the informative prior in learning the new graph:

$$NE = \max_i \frac{ne_{i,new}}{ne_{new}} \tag{2.18}$$

where $ne_{i,new}$ is the number of nodes that have the same parent nodes between the ith historical graph and the new graph, ne_{new} is the number of nodes in the new graph. The larger the NE is, the higher power the informative prior can obtain. For example, in Fig.

2.7, we have $NE = \max(\frac{9}{15}, \frac{7}{15}) = 0.6$. In the simulations, we randomly rearrange the edges in the new graphs so that we get different new processes with different values of NE. To evaluate the learning results, we use the percentage of correctly identified edges [33] as the index to indicate the effectiveness of two methods. We average 100 times structure learning results when using different number of observation data. All the scaling parameter λs introduced in Equation (2.11) are assumed to follow an uniform distribution between (0,1000). The new graph structure learning results for different NEs are shown in Fig. 2.8. We can see that with the help of the informative prior, it is more accurate in the graph structure learning. Even if we mismatch 30% of edges among the new graph and historical graphs, the learning result still outperforms the traditional non-informative prior metric. When we have most of the edges in the historical graphs mismatched in the new graph, we can still learn the graph as effective as the non-informative prior case due to the correcting algorithm we adopted. Thus, the simulation studies show the superiority and effectiveness of the proposed informative prior construction and graph learning methods. Furthermore, we investigate the influence of λ on the effectiveness on the proposed method. The results of different ranges of λ are shown in Fig. 2.9, where we take the $NE = 80\%$ case for illustration. We can see that as the integration range of λ becomes smaller, the proposed algorithm gradually degenerates to the non-informative Bayesian score. This is consistent with the analysis in Equation (2.12) that as $\lambda \to 0$ the edge existence probability becomes $\alpha = 0.5$, which is independent with the historical data.

2.6 Case study

We apply the proposed methodology to the car body assembly process shown in the Fig. 2.10. The final product of the car body assembly process is shown in Fig. 2.10(a). Fig. 2.10(b) is a simplified assembly process, where a set of sequenced assemblies, e.g., dash,

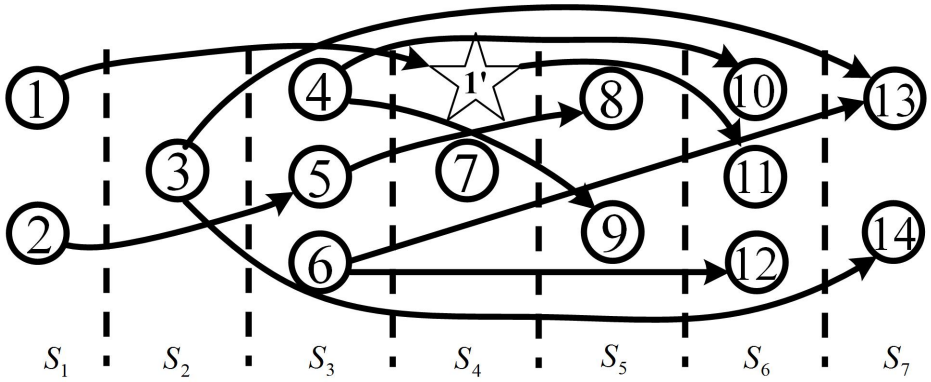


Figure 2.7: One possible new production line in the manufacturing system.

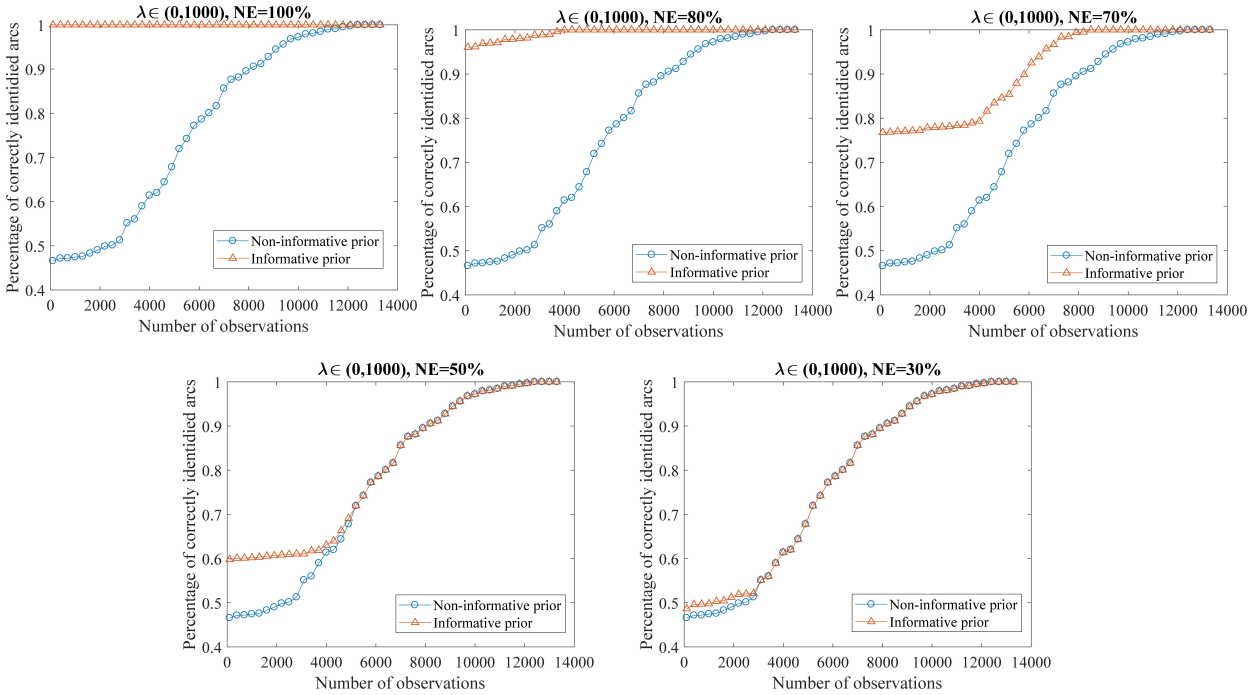


Figure 2.8: The structure learning performance for new production line.

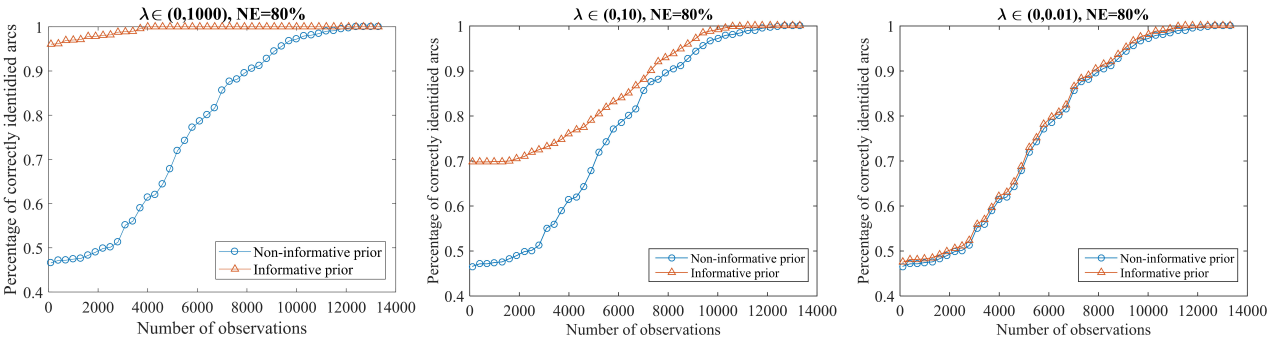


Figure 2.9: The influence of λ on the structure learning performance.

body sides, roof, are presented. Due to the complexity of the process, it is important to identify the mutual influences among key product characteristics (KPCs). To simplify the analysis and show the effectiveness of the proposed method, we choose 14 KPCs to validate the proposed method. These KPCs are random variables that denote the position deviations from the nominal value in the corresponding features. The physical layout of these KPCs is illustrated in Fig. 2.11(a). The assembly line for different products have different KPCs, and two historical assembly lines with selected KPCs are shown in Fig. 2.11(b). The new assembly line's KPC layout is in Fig. 2.12, which contains different KPCs with historical lines. Our task is to identify the mutual relationship among KPCs in the new line with the help of two historical lines. The assembly process is simulated in 3DCS, a commercially used dimensional variation simulation software for assembly and machining processes. This software is based on the governing physical laws in joining to simulate the assembly and is widely used in practice for dimension management. The KPCs in different lines can be obtained through the outputs of 3DCS. The normality and linear interaction assumptions of the data are validated in the work [34, 35]. We go through the steps in Section 2.4 and assign $\alpha = 0.5$ and $\lambda \in (0, 1000)$. The learned historical graphs are shown in Fig. 2.13. We construct the Bayesian score with the informative prior

and the final learned graph is shown in Fig. 2.14, where we use only 31 observation data points. The non-informative Bayesian score can also converge to the graph in Fig. 2.14, but with 67 observation data points in each node.

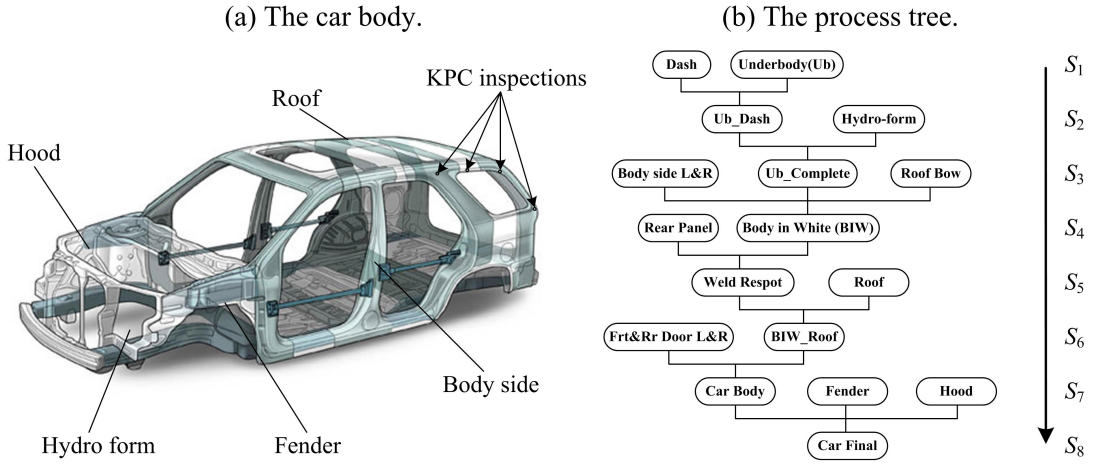


Figure 2.10: The car body assembly process.

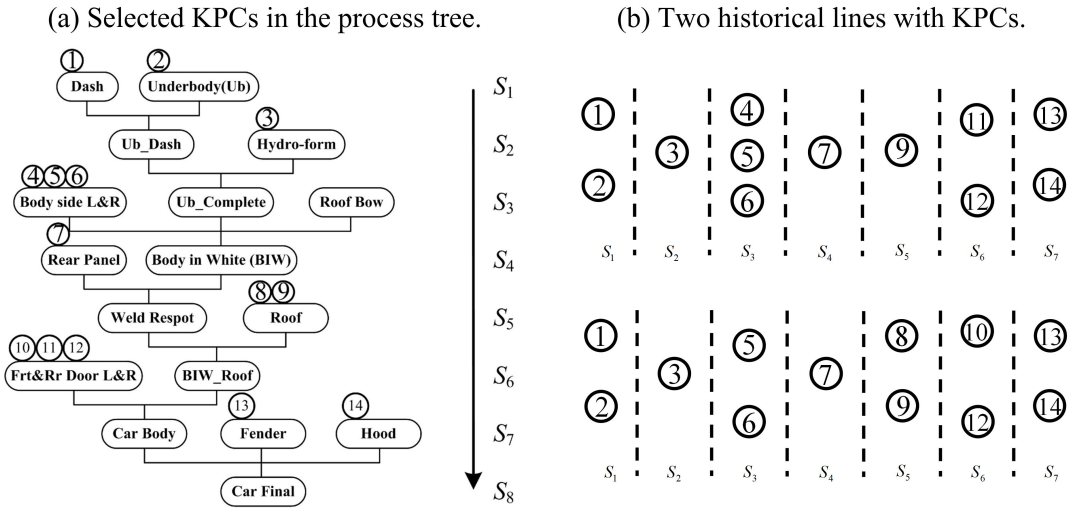


Figure 2.11: Physical layout for the assembly process in the case study.

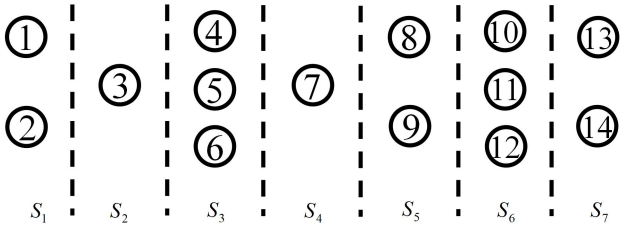


Figure 2.12: The new assembly line layouts.

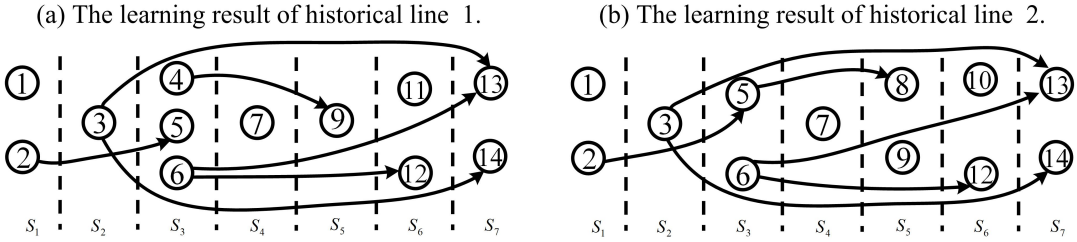


Figure 2.13: The learning results for historical lines.

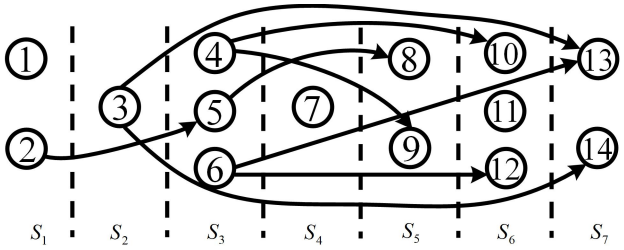


Figure 2.14: The final learning result for the assembly process.

We use the method proposed in [36] to validate the structure learning result for the case study. The general idea in [36] is to re-sample the data using bootstrapping method, and learn the graph structure using the re-sample data to check the robustness of the learning result. In our work, we use the non-parametric bootstrapping to re-sample the case study data 100 times, and the learning result presented in Fig. 2.14 appears 94 times among the re-sample based structure learning, which validates that the learning result in Fig. 2.14 fits the data best.

It is also verified the identified interactions among nodes can be interpreted by their

physical relationships. For example, nodes 4, 9 and 10 represent the deviations of points on the flush surface of the right body side (x_4), the right rear edge of the roof (x_9), and the outer surface of the rear door (x_{10}), respectively. The relationships diagram for these three nodes in the car body assembly process are shown in Fig. 2.15. The body side will be first welded on the underbody, then the roof will be added by fixtures located on the body side. The rear door installation is also related to the hinges attached on the body side. Thus, the body side decides the positions of the roof and the rear door, which justifies the identified relationship from node 4 to node 9 and 10. It is also clear that node 9 and node 10 do not interaction with each other, which is also reflected in the identified graph in Fig. 2.14. This case study verifies the effectiveness and superiority of the proposed informative prior based graph structure learning method.

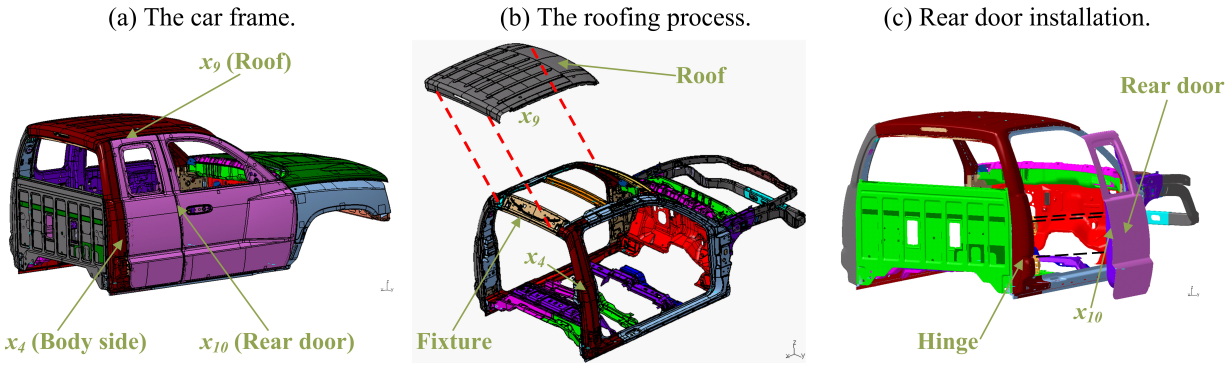


Figure 2.15: Interpretations of two identified direct influences.

2.7 Conclusion and Discussion

In this chapter, we propose a structure learning method for ordered block graphical models. This is a score based method and mainly uses the Bayesian score framework that consists of the likelihood term and the structure prior term. The key innovation in this

method is that we establish an informative structure prior by merging different historical graphs into the structure prior term under the Bayesian framework. The informative structure prior is expected to use historical information to benefit new graph structure learning. The key challenge of knowledge transferring in this work is the inconsistency of the sample space between historical processes and the new process. We propose a strategy by marginalizing inconsistent nodes and adding pseudo nodes to make the sample space consistent. A correcting algorithm based on the Bayes factor is also proposed to deal with the non-conformity between the prior knowledge and the new graph. We investigate the robustness of the proposed method and find that the proposed method performs better than the method using non-informative priors in most practical cases and no worse than the non-informative prior based method in general. The proposed method is also verified through a car body assembly process, where we successfully identify the relationships among assembly stations with much less data than that in the traditional method.

We would like to point out that although our work focuses on the OBM based structure learning, it can be extended to the structure learning of general graphical models. The idea is to use the "ordering based search" [18] to first order the nodes according to prior knowledge (or randomly order), then use our OBM method to learn the ordered graph. According to the learning result, we can cache the learning results and re-order all the nodes and repeat the OBM learning method. As the re-ordering proceeds, lots of ordered sub-graphs have already been learned and cached in the previous calculation, which offers computation efficiency. As a result, our algorithm can be integrated with existing searching algorithm to learn the general graphical models. How to leverage on the ordering based search method with the proposed method in general graphical model learning is the problem we will study in the future.

Another point we would like to mention is the calculation issue in Equation (2.6),

where we need to go through $|\mathbf{\Omega}_{\mathcal{G}_k}|^n$ different sub-graph realizations and sum over them. Although Equation (2.6) provides a rigorous way to infer the sub-graph distribution, the exponentially increasing calculation load makes it feasible only for small number of historical graphs. Alternatively, when we need to deal with large amount of historical graphs, the Markov Chain Monte Carlo (MCMC) method, e.g., Gibbs sampling, can be used to numerically approximate the predictive distribution. The details about MCMC based approximation can be found in [26]. Another practical issue is the selection of the sparse constrain of d when we do not have prior knowledge about d. In this case, we can resort to the unique identification property of the OBM, which says the structure learning of OBM will have an unique solution if sufficient data is given [4]. In practice, we may only have limited data and this result may not hold strictly. Nevertheless, we can use this property to develop a simple heuristic procedure to select d: we can first set a small d, say 2 or 3 and conduct the proposed graphical structure learning. Then we gradually increase d until the learning result does not change.

Some other interesting open issues remain in the proposed method. For example, In general, avoiding "negative" transfer of information, i.e., removing the historical data that does not help the learning, is an important yet challenging problem in transfer learning. The method of using the Bayes factor to correct the graphs in the post-analysis step is a primitive way of avoiding "negative" transfer. It will be interesting to develop more sophisticated method to avoid "negative" transfer during the learning, rather than post-learning. Another open issue is the multi-normal assumption. In our work, the normality assumption offers great convenience in modeling the graphical model learning as a conditional regression framework. However, this assumption may be violated in some cases, especially when the dataset is small. Some kernel based testing approaches [37] have been proposed to relax the normality assumption in conditional independence testing.

How to efficiently combine these methods and ideas into the graphical structure learning problem is another interesting yet challenging task to be studied in the future.

2.8 Appendix

2.8.1 Derivation of $P(\mathcal{G}_k = g_{k,s^*}|\mathbf{D}^1, \cdots, \mathbf{D}^n)$

We first deal with the general case for our problem, and then present our result shown in the chapter.

Suppose $\mathcal{G} \sim Cat(p_1, \cdots, p_\mathcal{K})$ and $(P_1, \cdots, P_\mathcal{K}) \sim Dir(\xi_1, \cdots, \xi_\mathcal{K})$. We have n i.i.d samples from $\mathcal{G}$ and the observed data $\mathbf{D}^i$ are samples from $\mathbf{Y} \sim N(\mathbf{0}, \mathbf{\Sigma}_i)$, where $\mathbf{\Sigma}_i$ represents the co-variance matrix depends on the ith sample from $\mathcal{G}$, $i = 1, \cdots, n$. Then we can estimate the event probabilities p_{g^*}, $g^* \in \{1, \cdots, \mathcal{K}\}$ as follows:

$$
\begin{aligned}
&P(\mathcal{G} = g^*|\mathbf{D}^1, \cdots, \mathbf{D}^n) \\
&= \sum_{\boldsymbol{g}} P(\mathcal{G} = g^*|\mathbf{D}^1, \cdots, \mathbf{D}^n, \boldsymbol{g}) P(\boldsymbol{\mathcal{G}} = \boldsymbol{g}, \mathbf{D}^1, \cdots, \mathbf{D}^n) \\
&= \sum_{\boldsymbol{g}} P(\mathcal{G} = g^*|\boldsymbol{g}) P(\mathbf{D}^1, \cdots, \mathbf{D}^n|\boldsymbol{\mathcal{G}} = \boldsymbol{g}) P(\boldsymbol{\mathcal{G}} = \boldsymbol{g}) \\
&= \sum_{\boldsymbol{g}} P(\mathcal{G} = g^*, \boldsymbol{g}) \prod_{i=1}^{n} P(\mathbf{D}^i|\mathcal{G} = g^i) \\
&\propto \sum_{\boldsymbol{g}} P(\mathcal{G} = g^*, \boldsymbol{g}) \prod_{i=1}^{n} \frac{P(\mathcal{G} = g^i|\mathbf{D}^i)}{P(\mathcal{G} = g^i)}
\end{aligned}
\tag{2.19}
$$

where $\boldsymbol{g} = \{g^1, \cdots, g^n\}$ is a vectored set of possible samples from the categorical distribution for sampling n times, and $g^i \in \{1, \cdots, \mathcal{K}\}$, $i = 1, \cdots, n$.

Consider the g^* and $\boldsymbol{g}$ are i.i.d. samples from the categorical distribution, and the event probabilities of the categorical distribution follow the Dirichlet distribution, we have

the $P(\mathcal{G} = g^*, \boldsymbol{g})$as follows for every possible $\boldsymbol{g}$:

$$P(\mathcal{G} = g^*, \boldsymbol{g}) = \int_{\boldsymbol{p}} f(\boldsymbol{p}) \prod_{l=1}^{\mathcal{K}} p_l^{c_l} \mathrm{d}\boldsymbol{p} \tag{2.20}$$

where $f(\boldsymbol{p})$ is the probability density function of the Dirichlet distribution and c_l is the number of samples that equals to j, e.g., $c_l = \sum_{i=1}^{n} \mathbf{1}(g^i = l) + \mathbf{1}(g^* = l)$ with $g^i \in \boldsymbol{g}$ and $\mathbf{1}(\cdot)$ is the indicator function. Furthermore, Equation (2.20) can be treated as the expectation of $\prod_{l=1}^{\mathcal{K}} p_l^{c_l}$ over the $\boldsymbol{p}$, and we have:

$$\int_{\boldsymbol{p}} f(\boldsymbol{p}) \prod_{l=1}^{\mathcal{K}} p_l^{c_l} \mathrm{d}\boldsymbol{p} = \mathbb{E}(\prod_{l=1}^{\mathcal{K}} p_l^{c_l}) = \frac{\Gamma(\sum_{l=1}^{\mathcal{K}})}{\Gamma\Big(\sum_{l=1}^{\mathcal{K}}(\xi_l + c_l)\Big)} \prod_{l=1}^{\mathcal{K}} \frac{\Gamma(\xi_l + c_l)}{\Gamma(\xi_l)} \tag{2.21}$$

Thus, we have:

$$P(\mathcal{G} = g^* | \mathbf{D}^1, \cdots, \mathbf{D}^n) \propto \sum_{\boldsymbol{g}} \Big\{ \frac{\Gamma(\sum_{l=1}^{\mathcal{K}} \xi_l)}{\Gamma\Big(\sum_{l=1}^{\mathcal{K}}(\xi_l + c_l)\Big)} \prod_{l=1}^{\mathcal{K}} \frac{\Gamma(\xi_l + c_l)}{\Gamma(\xi_l)} \prod_{i=1}^{n} P(\mathcal{G} = g^i | \mathbf{D}^i) \Big\} \tag{2.22}$$

By using the flat Dirichlet distribution that uniformly assigns the density of event probabilities, e.g.,$\xi_1 = \cdots = \xi_{|\boldsymbol{\Omega}_{\mathcal{G}_k}|} = 1$, we can have the result as:

$$P(\mathcal{G} = g^* | \mathbf{D}^1, \cdots, \mathbf{D}^n) \propto \sum_{\boldsymbol{g}} \Big\{ \frac{\Gamma(\mathcal{K})}{\Gamma\Big(\sum_{l=1}^{\mathcal{K}}(c_l) + \mathcal{K}\Big)} \prod_{l=1}^{\mathcal{K}} \frac{\Gamma(1 + c_l)}{1} \prod_{i=1}^{n} P(\mathcal{G} = g^i | \mathbf{D}^i) \Big\} \tag{2.23}$$

Applying Equation (2.23) to our problem by replacing the categorical random variable $\mathcal{G}$ as $\mathcal{G}_k$. Also note that the index vector $\boldsymbol{g}$ has $|\boldsymbol{\Omega}_{\mathcal{G}_k}|^n$ different combinations, which can be indexed by $S_{(i,j)}, i = 1, \cdots, n, j = 1, \cdots, |\boldsymbol{\Omega}_{\mathcal{G}_k}|$. As a result, we have:

$$P(\mathcal{G}_k = g_{k,s^*}|\mathbf{D}^1,\cdots,\mathbf{D}^n) \propto \sum_{j=1}^{|\mathbf{\Omega}_{\mathcal{G}_k}|^n} \left\{ \frac{\Gamma(|\mathbf{\Omega}_{\mathcal{G}_k}|)}{\Gamma(\sum_{l=1}^{|\mathbf{\Omega}_{\mathcal{G}_k}|} c_l + |\mathbf{\Omega}_{\mathcal{G}_k}|)} \prod_{l=1}^{|\mathbf{\Omega}_{\mathcal{G}_k}|} \big(\Gamma(1+c_l)\big) \cdot \prod_{i=1}^{n} P(\mathcal{G}_k^i = g_{k,S(i,j)}|\mathbf{D}^i) \right\} \tag{2.24}$$

2.8.2 Proof of lemma 1

Let Q, W, T and Z be random vectors. The following properties of conditional independence will be used:

P1 $Q \perp\!\!\!\perp (W,T)|Z \Rightarrow Q \perp\!\!\!\perp W|Z$ and $Q \perp\!\!\!\perp T|Z$.

P2 $Q \perp\!\!\!\perp W|Z$ and $Q \perp\!\!\!\perp T|Z \Rightarrow Q \perp\!\!\!\perp (W,T)|Z$.

Property P1 historical whenever Q, W, T and Z have a positive joint density with respect to a product measure. P2 is true only when Q, W, T and Z are jointly normal.

The Equation (7) in the manuscript can be reformulated as follows:

$$\begin{aligned} P(\mathcal{G}_k^i = g_{k,s}|\mathbf{D}^i) &= P(\mathbf{E}_{k,s} = \mathbf{1}, \bar{\mathbf{E}}_{k,s} = \mathbf{0}|\mathbf{D}^i) \\ &= P(\mathbf{E}_{k,s} = \mathbf{1}|\bar{\mathbf{E}}_{k,s} = \mathbf{0}, \mathbf{D}^i) P(\bar{\mathbf{E}}_{k,s} = \mathbf{0}|\mathbf{D}^i) \end{aligned} \tag{2.25}$$

The first part in Equation (2.25) can be reformulated based on the chain rule:

$$\begin{aligned} &P(\mathbf{E}_{k,s} = \mathbf{1}|\bar{\mathbf{E}}_{k,s} = \mathbf{0}, \mathbf{D}^i) \\ &= P(E^1_{\wp_{k,s}} = 1|\bar{\mathbf{E}}_{k,s} = \mathbf{0}, \mathbf{D}^i) \cdots P(E^l_{\wp_{k,s}} = 1|E^1_{\wp_{k,s}} = 1, \cdots, E^{l-1}_{\wp_{k,s}} = 1, \bar{\mathbf{E}}_{k,s} = \mathbf{0}, \mathbf{D}^i) \\ &\propto \prod_{w=1}^{l} P(E^w_{\wp_{k,s}} = 1|\bar{\mathbf{E}}_{k,s} = \mathbf{0}, \mathbf{D}^i) \end{aligned} \tag{2.26}$$

The second part in Equation (2.25) represents tests for the conditional independencies, we define $\wp_{k,s} = [Y^1_{\wp_{k,s}}, \cdots, Y^l_{\wp_{k,s}}]$ and $\bar{\wp}_{k,s} = [Y^1_{\bar{\wp}_{k,s}}, \cdots, Y^r_{\bar{\wp}_{k,s}}]$ as the parent nodes and non-parent nodes for the node $Y_{k,s}$, then we can have:

$$\bar{\mathbf{E}}_{k,s} = \mathbf{0}|\mathbf{D}^i \equiv Y_{k,s} \perp\!\!\!\perp (Y^1_{\bar{\wp}_{k,s}}, \cdots, Y^r_{\bar{\wp}_{k,s}}) | Y^1_{\wp_{k,s}}, \cdots, Y^l_{\wp_{k,s}} \tag{2.27}$$

With the properties P1 and P2 and the assumption A1, we can have:

$$\begin{aligned} & Y_{k,s} \perp\!\!\!\perp (Y^1_{\bar{\wp}_{k,s}}, \cdots, Y^r_{\bar{\wp}_{k,s}}) | Y^1_{\wp_{k,s}}, \cdots, Y^l_{\wp_{k,s}} \\ & \equiv (Y_{k,s} \perp\!\!\!\perp Y^1_{\bar{\wp}_{k,s}} | Y^1_{\wp_{k,s}}, \cdots, Y^l_{\wp_{k,s}}), \cdots, (Y_{k,s} \perp\!\!\!\perp Y^r_{\bar{\wp}_{k,s}} | Y^1_{\wp_{k,s}}, \cdots, Y^l_{\wp_{k,s}}) \\ & \equiv (E^1_{\bar{\wp}_{k,s}} = 0 | \bar{\mathbf{E}}^{-1}_{k,s} = \mathbf{0}, \mathbf{D}^i), \cdots, (E^r_{\bar{\wp}_{k,s}} = 0 | \bar{\mathbf{E}}^{-r}_{k,s} = \mathbf{0}, \mathbf{D}^i) \end{aligned} \tag{2.28}$$

Then we can have the $P(\bar{\mathbf{E}}_{k,s} = \mathbf{0}|\mathbf{D}^i)$ as follows:

$$P(\bar{\mathbf{E}}_{k,s} = \mathbf{0}|\mathbf{D}^i) \propto \prod_{u=1}^{r} P(E^u_{\bar{\wp}_{k,s}} = 0 | \bar{\mathbf{E}}^{-u}_{k,s} = \mathbf{0}, \mathbf{D}^i) \tag{2.29}$$

Plug Equation (2.26) and (2.29) into Equation (2.25), we can have:

$$\begin{aligned} & P(\mathcal{G}^i_k = g_{k,s} | \mathbf{D}^i) \\ & \propto \prod_{w=1}^{l} P(E^w_{\wp_{k,s}} = 1 | \bar{\mathbf{E}}_{k,s} = \mathbf{0}, \mathbf{D}^i) \cdot \prod_{u=1}^{r} \left(1 - P(E^u_{\bar{\wp}_{k,s}} = 1 | \bar{\mathbf{E}}^{-u}_{k,s} = \mathbf{0}, \mathbf{D}^i)\right) \end{aligned} \tag{2.30}$$

2.8.3 Derivation of $P(\mathbf{D}|\mathcal{G})$

Based on the Equation (15) in the manuscript and the assumtption A2, we can decompose the likelihood term:

$$P(\mathbf{D}|\mathcal{G}) = \prod_{k=1}^{q} P(\mathbf{Y}_k | g_{k,s^*}, \wp_{k,s^*}) \tag{2.31}$$

where $\mathbf{D} = \{\mathbf{Y}_1, \cdots, \mathbf{Y}_q\}$, $\mathbf{Y}_k = (Y_k^1, Y_k^2, \cdots, Y_k^v)^T$ represents the v samples of the random variable Y_k. Since the Y_k also depends on $\boldsymbol{\beta}_{k,s^*}$ and σ^2_{k,s^*}, we have:

$$
\begin{aligned}
&P(\mathbf{D}|\mathcal{G}) \\
&= \prod_{k=1}^{q} \int_{\boldsymbol{\beta},\sigma^2|g_{k,s^*},\wp_{k,s^*}} \prod_{m=1}^{v} P(Y_k^m|g_{k,s^*},\wp_{k,s^*},\boldsymbol{\beta}_{k,s^*},\sigma^2_{k,s^*}) \cdot P(\boldsymbol{\beta},\sigma^2|g_{k,s^*},\wp_{k,s^*})\mathrm{d}\boldsymbol{\beta},\sigma^2|g_{k,s^*},\wp_{k,s^*} \\
&= \prod_{k=1}^{q} \iint_{\boldsymbol{\beta}_{k,s^*},\sigma^2_{k,s^*}} \prod_{m=1}^{v} P(Y_k^m|\boldsymbol{\beta}_{k,s^*},\sigma^2_{k,s^*}) \cdot P(\boldsymbol{\beta}_{k,s^*},\sigma^2_{k,s^*})\mathrm{d}\boldsymbol{\beta}_{k,s^*}\mathrm{d}\sigma^2_{k,s^*}
\end{aligned}
\tag{2.32}
$$

We further assume $\boldsymbol{\beta}_{k,s^*}$ and σ^2_{k,s^*} jointly follow the Normal-inverse-Gamma distribution, then the $P(\boldsymbol{\beta}_{k,s^*},\sigma^2_{k,s^*})$ will be the conjugate prior for $P(Y_k^m|\boldsymbol{\beta}_{k,s^*},\sigma^2_{k,s^*})$ and gives:

$$
P(\boldsymbol{\beta}_{k,s^*},\sigma^2_{k,s^*}) = P(\boldsymbol{\beta}_{k,s^*}|\sigma^2_{k,s^*})P(\sigma^2_{k,s^*}) = N(\boldsymbol{\mu}_{\beta_{k,s^*}},\sigma_{k,s^*}\mathbf{V}_{\beta_{k,s^*}}) \cdot IG(\gamma_{k,s^*},\eta_{k,s^*}) \tag{2.33}
$$

where $\boldsymbol{\mu}_{\beta_{k,s^*}}$ and $\sigma_{k,s^*} \cdot \mathbf{V}_{\beta_{k,s^*}}$ are the mean vector and covariance matrix in the conditional normal distribution for $\boldsymbol{\beta}_{k,s^*}$, the γ_{k,s^*} and η_{k,s^*} are the parameters in the inverse Gamma distribution $IG(.,.)$ for σ^2_{k,s^*}. Define $c = |\wp_{k,s^*}|$ as the number of parent nodes. The derivation can be formulated as follows:

$$
\begin{aligned}
P(\mathbf{Y}_k) =& \iint_{\boldsymbol{\beta}_{k,s^*},\sigma^2_{k,s^*}} \prod_{m=1}^{v} P(Y^m_{k,s^*}|\boldsymbol{\beta}_{k,s^*},\sigma^2_{k,s^*})P(\boldsymbol{\beta}_{k,s^*},\sigma^2_{k,s^*})\mathrm{d}\boldsymbol{\beta}_{k,s^*}\mathrm{d}\sigma^2_{k,s^*} \\
=& \frac{\eta_{k,s^*}^{\gamma_{k,s^*}}}{(2\pi)^{\frac{c+1}{2}}\Gamma(\gamma_{k,s^*})\sqrt{|\mathbf{V}_{\beta_{k,s^*}}|}} \iint_{\boldsymbol{\beta}_{k,s^*},\sigma^2_{k,s^*}} \left(\frac{1}{\sigma^2_{k,s^*}}\right)^{\gamma_{k,s^*}+\frac{c+1+v}{2}+1} \cdot \\
& \exp\Big\{ -\frac{1}{\sigma^2_{k,s^*}}\Big[\eta_{k,s^*} + \frac{1}{2}\{(\boldsymbol{\beta}_{k,s^*} - \boldsymbol{\mu}_{\beta_{k,s^*}})^T\mathbf{V}_{\beta_{k,s^*}}(\boldsymbol{\beta}_{k,s^*} - \boldsymbol{\mu}_{\beta_{k,s^*}}) + \\
& (\mathbf{Y}_k - \tilde{\mathbf{Z}}_{\wp_{k,s^*}}\boldsymbol{\beta}_{k,s^*})^T(\mathbf{Y}_k - \tilde{\mathbf{Z}}_{\wp_{k,s^*}}\boldsymbol{\beta}_{k,s^*})\}\Big]\Big\}\mathrm{d}\boldsymbol{\beta}_{k,s^*}\mathrm{d}\sigma^2_{k,s^*}
\end{aligned}
\tag{2.34}
$$

The key to deriving the marginal distribution is the integration part, which requires the

multivariate completion of squares identity as follows:

$$\boldsymbol{\mu}^T A\boldsymbol{\mu} - 2\boldsymbol{\alpha}^T\boldsymbol{\mu} = (\boldsymbol{\mu} - A^{-1}\boldsymbol{\alpha})^T A(\boldsymbol{\mu} - A^{-1}\boldsymbol{\alpha}) - \boldsymbol{\alpha}^T A^{-1}\boldsymbol{\alpha} \tag{2.35}$$

Then Equation (2.34) can be reformulated as:

$$\begin{aligned}
&P(\mathbf{Y}_k)\\
=&\frac{\eta_{k,s^*}^{\gamma_{k,s^*}}}{(2\pi)^{\frac{c+1}{2}}\Gamma(\gamma_{k,s^*})\sqrt{|\mathbf{V}_{\beta_{k,s^*}}|}}\iint_{\beta_{k,s^*},\sigma^2_{k,s^*}}\left(\frac{1}{\sigma^2_{k,s^*}}\right)^{\gamma_{k,s^*}+\frac{c+1+v}{2}+1}\times\exp\Big\{-\frac{1}{\sigma^2_{k,s^*}}\big[\eta_{k,s^*}+\\
&\frac{1}{2}\{\boldsymbol{\mu}^T_{\beta_{k,s^*}}\mathbf{V}^{-1}_{\beta_{k,s^*}}\boldsymbol{\mu}_{\beta_{k,s^*}}+\mathbf{Y}_k^T\mathbf{Y}_k-{\boldsymbol{\mu}^*_{k,s^*}}^T{V^*_{k,s^*}}^{-1}\boldsymbol{\mu}^*_{k,s^*}\}+\\
&\frac{1}{2}(\boldsymbol{\beta}_{k,s^*}-\boldsymbol{\mu}^*_{k,s^*})^T{V^*_{k,s^*}}^{-1}(\boldsymbol{\beta}_{k,s^*}-\boldsymbol{\mu}^*_{k,s^*})\big]\Big\}\mathrm{d}\boldsymbol{\beta}_{k,s^*}\mathrm{d}\sigma^2_{k,s^*}\\
=&\frac{\eta_{k,s^*}^{\gamma_{k,s^*}}}{(2\pi)^{\frac{c+1+v}{2}}\Gamma(\gamma_{k,s^*})\sqrt{|\mathbf{V}_{\beta_{k,s^*}}|}}\\
&\times\frac{\Gamma(\gamma_{k,s^*}+\frac{v}{2})(2\pi)^{\frac{c+1}{2}}\sqrt{|V^*_{k,s^*}|}}{\Big(\eta_{k,s^*}+\frac{1}{2}\{\boldsymbol{\mu}^T_{\beta_{k,s^*}}\mathbf{V}^{-1}_{\beta_{k,s^*}}\boldsymbol{\mu}_{\beta_{k,s^*}}+\mathbf{Y}_k^T\mathbf{Y}_k-{\boldsymbol{\mu}^*_{k,s^*}}^T{V^*_{k,s^*}}^{-1}\boldsymbol{\mu}^*_{k,s^*}\}\Big)^{\gamma_{k,s^*}+\frac{v}{2}}}\\
=&\frac{\eta_{k,s^*}^{\gamma_{k,s^*}}\Gamma(\gamma_{k,s^*}+\frac{v}{2})\sqrt{|V^*_{k,s^*}|}}{(2\pi)^{\frac{v}{2}}\Gamma(\gamma_{k,s^*})\sqrt{|\mathbf{V}_{\beta_{k,s^*}}|}}\times\\
&\left[\eta_{k,s^*}+\frac{1}{2}\{\boldsymbol{\mu}^T_{\beta_{k,s^*}}\mathbf{V}^{-1}_{\beta_{k,s^*}}\boldsymbol{\mu}_{\beta_{k,s^*}}+\mathbf{Y}_k^T\mathbf{Y}_k-{\boldsymbol{\mu}^*_{k,s^*}}^T{V^*_{k,s^*}}^{-1}\boldsymbol{\mu}^*_{k,s^*}\}\right]^{-(\gamma_{k,s^*}+\frac{v}{2})}
\end{aligned} \tag{2.36}$$

where

$$\begin{aligned}
\boldsymbol{\mu}^*_{k,s^*}&=(\mathbf{V}^{-1}_{\beta_{k,s^*}}+\tilde{\mathbf{Z}}^T_{\wp_{k,s^*}}\tilde{\mathbf{Z}}_{\wp_{k,s^*}})^{-1}(\mathbf{V}^{-1}_{\beta_{k,s^*}}\boldsymbol{\mu}_{\beta_{k,s^*}}+\wp^T_{k,s^*}\mathbf{Y}_k)\\
V^*_{k,s^*}&=(\mathbf{V}^{-1}_{\beta_{k,s^*}}+\tilde{\mathbf{Z}}^T_{\wp_{k,s^*}}\tilde{\mathbf{Z}}_{\wp_{k,s^*}})^{-1}
\end{aligned} \tag{2.37}$$

where $\tilde{\mathbf{Z}}_{\wp_{k,s^*}}=\{\mathbf{Z}^1_{\wp_{k,s^*}},\cdots,\mathbf{Z}^v_{\wp_{k,s^*}}\}^T$ represents the v samples of the random variable set $\mathbf{Z}_{\wp_{k,s^*}}$ in $\mathcal{G}_{k,s^*}$.

3 APPROXIMATE KEY PERFORMANCE INDICATORS JOINT DISTRIBUTION THROUGH ORDERED BLOCKED MODEL AND PAIR COPULA CONSTRUCTION*

Key performance indicators (KPIs) play an important role in comprehending and improving the manufacturing system. This chapter focuses on quantifying non-Gaussian data under non-linear relationship and proposes a novel method using ordered block model and pair-copula construction (OBM-PCC) to approximate the multivariate distribution of KPIs. The KPIs are treated as random variables in the OBM and studied under the stochastic queuing framework. The dependence structure of the OBM represents the influence flow from system input parameters to KPIs. Based on the OBM structure, the PCC is employed to simultaneously approximate the joint probability density function represented by KPIs and quantify the KPI values. The OBM-PCC model removes the redundant pair-copulas in traditional modeling, at the same time enjoys the flexibility and desirable analytical properties in KPI modeling, thus efficiently providing the accurate approximation. Extensive numerical studies are presented to demonstrate the effectiveness of the OBM-PCC model.

3.1 Introduction

Key Performance Indicators (KPIs) are designed to measure critical system objectives and play an important role in performance evaluation and management. It is well documented that the prediction and quantification of KPIs relationship will lead to a better

*This chapter is based on the paper: **Wang C.** and Zhou S. (2018), "Approximate Key Performance Indicator Joint Distribution through Ordered Block Model and Pair Copula Construction", *IISE Transactions*, Accepted.

understanding of the system and benefit the manufacturing productivity improvement [38, 39, 40]. For example, redundant KPIs can be discovered and the continuous improvement can be implemented based on the predicted KPI and their relationships.

To study the manufacturing system performance and KPI, stochastic queuing system model is often used [41, 42]. A typical manufacturing process is illustrated as a queuing network in Fig. 1, where λ represents the arrival rate, c is the cycle time, μ_i and q_j are the service rate and queuing time for the server i and buffer j, respectively. Each buffer has a fixed buffer size $N_j \in \{1, 2, \cdots\}$. We define the input (λ, μ_i) of the process as system parameters and the output such as (c, q_j) as the response KPIs. The overall layout is shown in Fig. 1 (b), where l is the number of system input parameters and p is the total number of input parameters and response KPIs. For effective manufacturing system control, it is highly desirable to obtain the relationship among the system input parameters and the response KPIs. It is well known that for a queuing system, the output is stochastic for a given fixed input parameters even under stationary condition. Thus, to fully quantify the relationship between inputs and outputs, we need the multivariate distributions of response KPIs conditional on system input parameters, such as $f(q_1, c|\lambda, \mu_1)$ and $f(q_1, q_2, q_3|\lambda)$. With the multivariate conditional distributions, we can get mean, variance, quantile and their relationships under the interested system inputs to better understand the system. Here we would like to point out that the conditional distributions are referred to the stationary distributions of response KPIs.

There are some existing works on quantifying the relationship between system inputs and the response KPIs. However, most of existing methods cannot recover the multivariate conditional distribution of KPIs. For example, flexible simulation models were developed to quantify and predict process input-output values [43, 44]; different queuing models on batch arrivals [45] and mix products [46] were proposed to approximate the real

manufacturing system; data-driven based quantile regression for cycle time in the general queuing system was proposed to give estimation of cycle time under different arrival and service rate [47]. These methods are either time-consuming (simulation based) or restricted to explicit relationship among KPIs (regression based), thus cannot provide flexible and comprehensive picture of the KPI distributions. Non-parametric models, such as Kriging and spline, do not require explicit forms of the KPI relationships [39]. However, those non-parametric methods mainly focus on the relationship among the moments of KPIs (e.g., mean relationship). The comprehensive multivariate distribution of KPIs are not provided in those methods.

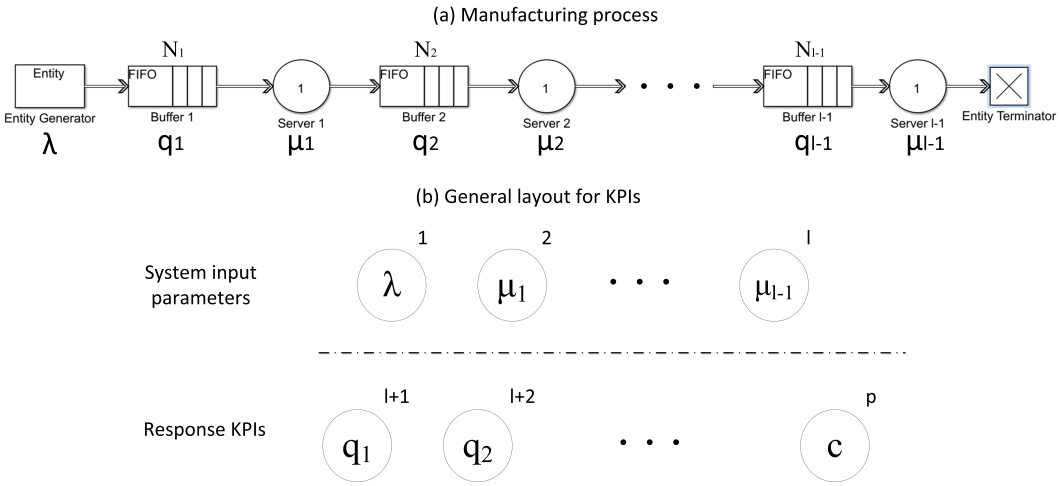


Figure 3.1: KPIs in a manufacturing process.

In order to obtain the multivariate distributions of an arbitrary combination of KPIs conditioning on a wide range of system input parameters, we propose a novel approximation method in this chapter. The basic idea is that we treat both the response KPIs and the system input parameters as random variables, and fit the joint probability density function (pdf) among them, e.g., $f(\lambda, \mu_1, \cdots, \mu_{l-1}, q_1, \cdots, c)$, using the simulated system input-output data under stationary condition. Then, an arbitrary conditional pdfs can be obtained through sampling strategies, e.g., Markov Chain Monte Carlo, from the joint

pdf. In more details, in the first step, we can generate a set of training samples from the assumed joint distribution, e.g., $f(\lambda, \mu_1, \cdots, \mu_{l-1}, q_1, \cdots, c)$, through simulation. Without the loss of generality, we can specify the system input parameters to follow non-informative uniform distributions, where the range of the distribution is based on the interested range of the input parameters. We sample from the specified uniform distributions and then use the sampled input parameters to simulate the production system. Once the system is in the stationary state, we pick the system response as a sample of KPIs corresponding to the sampled input parameters. We then repeat this process multiple times to obtain a training dataset with samples from the joint distribution $f(\lambda, \mu_1, \cdots, \mu_{l-1}, q_1, \cdots, c)$. In the second step, we need to fit the joint distribution using the training samples. This is an important step because with the fitted joint distribution, we can easily obtain the conditional distribution of KPIs under an arbitrary input parameter setting without conducting tedious and time consuming system simulation.

Fitting high dimensional multivariate distribution function is in general a challenging problem. Some methods have been proposed [48, 49, 50, 51, 52, 53]. These methods can be roughly classified as parametric methods [48, 49, 50] and non-parametric methods [51, 52, 53]. In practice, the distribution of KPIs and their relations for a production system is often very complex. In this chapter, we propose to apply copula analysis to approximate the joint pdf and quantify the relationship among KPIs. Copula is a multivariate probability distribution with standard uniform univariate margins, which links the marginal distribution and the joint distribution in the multivariate analysis [54]. The pair-copula construction (PCC) first proposed in [55] is a hierarchical extension for the copulas through the construction of different pair-wise bi-variate copulas. The PCC is widely used in the multivariate dependence quantification in financial and risk management area [56, 57] due to its flexibility and easy-to-get analytic form. PCC also provides great

potential in approximating the complex joint pdf for multivariate random variables. One specific challenge in applying PCC to a high dimensional distribution fitting is how to refine and estimate concise pair-copulas from large amount of candidates. To address this challenge, we will take advantage of the prior knowledge of the dependence structure among the variables in our problem. Such knowledge can help to remove redundant PCCs and improve the PCC efficiency. For a production system that is modeled as a queuing system, the dependence structure among variables can be obtained through queuing theory or data-driven statistical tests [37]. For example, in the M/M/1 queuing, the arrival rate and service rate (system input parameters) determine the queuing time and the cycle time (response KPIs). As a result, the M/M/1 queuing has the dependence structure in Fig. 2, where only the system input parameters can influence on the response variables, which is represented by directed edges in the figure. The relationship in Fig. 2 gives a graphical representation of the dependence structure among variables, which can be formally described as an ordered block model (OBM) [5]. As introduced in Chapter 2, the OBM is a special form of graphical models, where the order of nodes representing the random variables in the OBM is assumed to be known. The inputs and outputs of queuing systems can be put into an OBM form. By combining OBM with PCC approach, we can establish an effective multivariate joint distribution approximation method for KPIs in production systems. The proposed method takes advantages of both domain knowledge for the queuing system and the PCC flexibility in approximating the joint distribution. Furthermore, parameter estimation can be computed recursively and efficiently due to the analytic form of PCC.

The rest of the chapter is organized as follows: Section 3.2 gives the detailed problem formulation and presents the proposed multivariate distribution approximation method for KPIs. Simulation experiments will be conducted in Section 3.3 to show the effectiveness

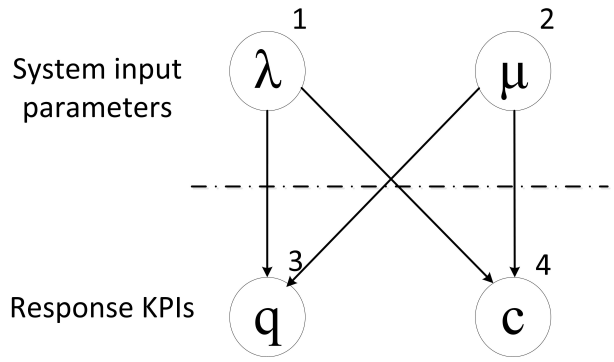


Figure 3.2: Dependence structure in the M/M/1 queue.

of the proposed method. Section 3.4 draws conclusion remarks.

3.2 Problem formulation and KPI quantification

3.2.1 Problem formulation and assumptions

In this work, we model the production system as a queuing system. We use variables $X_1, ..., X_p$ to represent the system input parameters and response KPIs. Specifically, the arrival rate λ and the service rate μ at different servers are considered as the input parameters, and the queuing time q for each server and the cycle time c are chosen as the response KPI variables. Please note that the proposed method is general for continuous random variables. Other continuous responses such as server utilizations can also be included as response KPIs. We further define $\{X_i : i \in \mathbf{I}\}$, $\mathbf{I} = \{1, ..., l\}$ and $\{X_j, j \in \mathbf{O}\}$, $\mathbf{O} = \{l+1, ..., p\}$ as the input parameters and response KPIs, respectively. Figure 1 (b) illustrates these variables, where $X_1 = \lambda, X_2 = \mu_1, ..., X_{l+1} = q_1$, ..., $X_p = c$. Here we give two assumptions on the process:

A4 The process $\{X_j^k, j \in \mathbf{O}\}$ is stationary for a given set $\{X_i = x_i, i \in \mathbf{I}\}$, and $X_j^k, j \in \mathbf{O}$ converges in distribution to the corresponding random

variable $X_j, j \in \mathbf{O}$ for the given set $\{X_i = x_i, i \in \mathbf{I}\}$ as $k \to \infty$, where k is the observation or the sample index.

A5 The process $\{X_j^k, j \in \mathbf{O}\}$ is ϕ-mixing for a given set $\{X_i = x_i, i \in \mathbf{I}\}$. In other words, let $M_{-\infty}^{e}$ and M_{e+d}^{∞} be σ-fields generated by $\{X_j^k, j \in \mathbf{O}; k \leq e\}$ and $\{X_j^k, j \in \mathbf{O}; k \geq e+d\}$ given $\{X_i = x_i, i \in \mathbf{I}\}$, if $D_1 \in M_{-\infty}^{e}$ and $D_2 \in M_{e+d}^{\infty}$, then

$$|P(D_2|D_1) - P(D_2)| \leq \phi(d), \quad \forall e, \tag{3.1}$$

where the σ-field serves as the collection of subsets for which the corresponding probability measure $P(\cdot)$ is defined [58], $e \in \mathbb{Z}$, $d \in \mathbb{N}^+$ and $\phi(d)$ is strictly non-increasing for $d \geq 1$, and $\lim_{d\to\infty} \phi(d) = 0$ [59].

These two assumptions are commonly adopted in queuing analysis [47, 60]. It is worth mentioning that although A4 is difficult to be formally validated in general situations, it is fairly intuitive and not restrictive to many applications. A5 imposes the dependence on the consecutive response observations, and assumes the dependence between two samples decreases as the samples become farther apart from each other. Therefore, if we take samples sufficiently far apart from each other in the steady state as the observations, they can be regarded as independent and identically distributed (i.i.d.) samples according to A4 and A5.

We use OBM to model the dependence structure among the variables in a queuing system. Let $G(V, E)$ be an OBM based on a set of nodes V and a set of directed edges E. Each variable $X_1, ..., X_p$ has a corresponding node in the $V = \{1, ..., p\}$. In the rest of the chapter, we will use nodes and random variables interchangeably. The directed edge E is an indicator variable that represents the influence from the parent node variable(s)

($\boldsymbol{X}_{pa(i)}$) to the descendant node variable (X_i) conditioning on all other nodes. For example, we have $\boldsymbol{X}_{pa(3)} = \{X_1, X_2\}$ in Fig. 2, which means nodes λ and μ have direct influence on the node q conditioning on other nodes. One of the important features of the OBM is that it can factorize the joint probability density function of $(X_1, ..., X_p)$ in terms of the parent/descendant relationship as follows:

$$f(x_1, ..., x_p) = \prod_{i=1}^{p} f_{i|pa(i)}\Big(x_i|\boldsymbol{x}_{pa(i)}\Big). \tag{3.2}$$

where the conditioning is on $\boldsymbol{x}_{pa(i)}$, the set of parent variables of X_i. We further define $\boldsymbol{X}_{nd(i)}$ as the set of non-descendants of the node X_i. Then we say the joint density function of $(X_1, ..., X_p)$ has the local Markov property with respect to G if

$$X_i \perp\!\!\!\perp \boldsymbol{X}_{nd(i)\backslash pa(i)} | \boldsymbol{X}_{pa(i)}, \quad \forall i \in V, \tag{3.3}$$

where we use symbol '\' to represent excluding element(s) from a set. If the conditional independence holds, then there is no edge between X_i and $\boldsymbol{X}_{nd(i)\backslash pa(i)}$, and vice versa.

The system input parameters and response KPIs of a queuing system construct a two stage OBM as shown in Fig. 1 (b). In this chapter, we assume the dependence relationship between the elements in these two stages are known. For a simple queuing system, the dependence relationship can be easily inferred from queuing theory. However, for a complex queuing network, the relationship may not be obvious. In such case, statistical testing methods can be applied to identify the edges in an OBM. For example, the kernel-based conditional independence test [37] can be used to identify the edges in an OBM for the variables in a queuing network, where most variables follows non-Gaussian distribution.

Based on the OBM structure, we want to approximate the joint pdf and quantify the response KPIs given specific system input parameters. A brief review of PCC will be given

in Section 3.2.2. Section 3.2.3 will build the link between OBM and PCC, and introduce the property of the proposed OBM-PCC method. The OBM-PCC based joint distribution approximation will be presented in Section 3.2.4. The overall summary of the proposed techniques is presented in Section 3.2.5.

3.2.2 Review of pair-copula construction

The copula is a multivariate cumulative distribution function (cdf) $C : [0,1]^p \to [0,1], p \in \mathbb{N}$ such that all the univariate marginals are uniform distributions on the interval $[0,1]$. The Sklar's theorem [61] shows that every cdf $F : \mathbb{R}^p \to [0,1]$ with univariate marginal $F_1, ..., F_p$ can be written as:

$$F(x_1, ..., x_p) = C\Big(F_1(x_1), ..., F_p(x_p)\Big), \tag{3.4}$$

for some copula C and all $x_1, ..., x_p \in \mathbb{R}$. If F is absolutely continuous and $F_1, ..., F_p$ are strictly increasing, we can transform Equation (5.8) to the pdf expression as:

$$f(x_1, ..., x_p) = c\Big(F_1(x_1), ..., F_p(x_p)\Big) \prod_{i=1}^{p} f_i(x_i), \tag{3.5}$$

where the copula pdf c is uniquely determined. The cdf C and pdf c of copula can be solved by marginal quantile functions. More specifically, we have

$$C(u_1, ..., u_p) = F\Big(F_1^{-1}(u_1), ..., F_l^{-1}(u_p)\Big), \tag{3.6}$$

$$c(u_1, ..., u_p) = \frac{f\Big(F_1^{-1}(u_1), ..., F_l^{-1}(u_p)\Big)}{\prod_{i=1}^{p} f_i\Big(F_i^{-1}(u_i)\Big)}, \tag{3.7}$$

for all $u_1, ..., u_p \in [0, 1]$. Equation (5.10) represents the multivariate copula, which is hard to model directly due to its complex tail behavior and heavy computation load. Joe, Bedford and Cooke [55, 62, 63] therefore proposed a flexible way to construct multivariate copulas through bi-variate copulas (pair-copula). The key idea of pair-copula construction is a graphical representation called 'regular vine' that consists of a sequence of trees, where each edge in the tree is associated with a certain pair-copula. We will briefly review the PCC using the example of D-vine, which is one of the widely applied regular vines.

Let F be the pdf of a D-vine PCC on $\mathbb{R}^p$ and let $V = \{1, ..., p\}$. The first row (or *tree*) in Fig. 5.2 of the D-vine comprises the p nodes that represent the univariate margins F_i of F. These nodes are connected by $p-1$ edges from node $i-1$ to i, $i \in V\backslash\{1\}$. These edges represent the unconditional pair-copulas $C_{(i-1),i}$. The subsequent trees of the D-vine are then derived from their predecessors by turning all edges into nodes and by introducing a new edge whenever two nodes share all but two indices. Those two indices form the conditioned set and the remaining ones the conditioning set of the associated pair-copula. The edges of the second tree, for example, denote the conditional pair-copula $C_{i,(i+2)|(i+1)}, i \in V\backslash\{p-1, p\}$. The D-vine with p nodes will have $p-1$ trees and $\binom{p}{2}$ edges. The pdf of F can be described in terms of pair-copula as follows [64]:

$$f(\mathbf{x}) = \prod_{i=1}^{p-1} \prod_{j=1}^{p-i} c_{j,j+i|(j,j+i)}\Big(F_{j|(j,j+i)}(x_j|\mathbf{x}_{(j,j+i)}), F_{j+i|(j,j+i)}(x_{j+i}|\mathbf{x}_{(j,j+i)})\Big) \prod_{k=1}^{p} f_k(x_k), \quad (3.8)$$

where we have $\mathbf{x}_{(j,j+i)} \triangleq (x_{j+1}, ..., x_{j+i-1})$ for all $i \leq p-1$ and $j \leq p-i$. We also denote the conditional cdf of X_j given $\mathbf{X}_{(j,j+i)} = \mathbf{x}_{(j,j+i)}$ by $F_{j|(j,j+i)}(x_j|\mathbf{x}_{(j,j+i)})$. Comparing Equation (5.12) and (5.9), we can see that the high dimension copula can be defined with $\binom{p}{2}$

(conditional) bi-variate copulas as follows:

$$c(u_1, \cdots, u_p) = \prod_{i=1}^{p-1} \prod_{j=1}^{p-i} c_{j,j+i|(j,j+i)}\Big(F_{j|(j,j+i)}(u_j|\mathbf{u}_{(j,j+i)}), F_{j+i|(j,j+i)}(u_{j+i}|\mathbf{u}_{(j,j+i)})\Big), \quad (3.9)$$

There are many studies on evaluating the stability of using bi-variate copulas to approximate the high dimension copula (joint distribution) [57, 65, 66]. The general conclusion is that the estimation of bi-variate copulas leads to stable performance in describing the high dimension copula (distribution) [65].

According to [55], the conditional cdfs $F_{j|K}, j \in V, K \subseteq V \backslash j$ can be computed using the recursive formula:

$$F_{j|K}(x_j|\mathbf{x}_K) = \frac{\partial C_{jk|(K\backslash\{k\})}\Big(F_{j|(K\backslash\{k\})}(x_j|\mathbf{x}_{(K\backslash\{k\})}), F_{k|(K\backslash\{k\})}(x_k|\mathbf{x}_{(K\backslash\{k\})})\Big)}{\partial F_{k|(K\backslash\{k\})}(x_k|\mathbf{x}_{(K\backslash\{k\})})}, \quad (3.10)$$

for $k \in K$. For instance, we can iteratively compute the values of $F_{j|(j,j+i)}$ and $F_{j+i|(j,j+i)}$ in tree i of the D-vine by choosing $k = j + i - 1$ and $k = j + 1$, respectively. Note that all the copulas needed in the current conditional cdf computation are already specified in preceding iterations. Thus, all the pair-copulas in a D-vine can be estimated recursively if we know the marginal cdf for each variable.

Take the four nodes D-vine in Fig. 5.2 for example, there are $\binom{4}{2} = 6$ edges (pair-copulas) for formulating the joint pdf as follows:

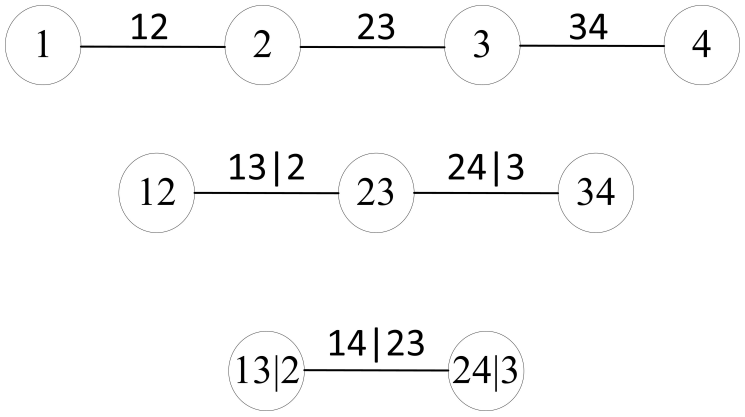


Figure 3.3: A four-variate D-vine.

$$
\begin{aligned}
&f(\mathbf{x}) \\
=&c_{12}\Big(F_1(x_1), F_2(x_2)\Big) \cdot c_{23}\Big(F_2(x_2), F_3(x_3)\Big) \cdot c_{34}\Big(F_3(x_3), F_4(x_4)\Big) \cdot c_{13|2}\Big(F_{1|2}(x_1|x_2), F_{3|2}(x_3|x_2)\Big) \\
&\cdot c_{24|3}\Big(F_{2|3}(x_2|x_3), F_{4|3}(x_4|x_3)\Big) \cdot c_{14|23}\Big(F_{1|23}(x_1|x_2,x_3), F_{4|23}(x_4|x_2,x_3)\Big) \prod_{i=1}^{4} f_i(x_i),
\end{aligned}
\tag{3.11}
$$

where the unconditional pair-copula $c_{12}\Big(F_1(x_1), F_2(x_2)\Big), c_{23}\Big(F_2(x_2), F_3(x_3)\Big)$ and $c_{34}\Big(F_3(x_3)$ $, F_4(x_4)\Big)$ can be obtained through the cdf $F_1(x_1), F_2(x_2), F_3(x_3)$ and $F_4(x_4)$. Then, the conditional cdf $F_{1|2}(x_1|x_2), F_{3|2}(x_3|x_2), F_{2|3}(x_2|x_3)$ and $F_{4|3}(x_4|x_3)$ can be achieved by Equation (5.13). For example, $F_{1|2}(x_1|x_2) = \frac{\partial C_{12}\Big(F_1(x_1),F_2(x_2)\Big)}{\partial F_2(x_2)}$, where the C_{12} has already been obtained. With these conditional cdfs, the conditional pair-copula $c_{13|2}\Big(F_{1|2}(x_1|x_2), F_{3|2}(x_3|x_2)\Big)$ and $c_{24|3}\Big(F_{2|3}(x_2|x_3), F_{4|3}(x_4|x_3)\Big)$ can be obtained. Similarly, by getting the conditional cdf $F_{1|23}(x_1|x_2,x_3)$ and $F_{4|23}(x_4|x_2,x_3)$ from Equation (5.13), the $c_{14|23}\Big(F_{1|23}(x_1|x_2,x_3)$, $F_{4|23}(x_4|x_2,x_3)\Big)$ is obtained. It is straightforward to see that getting the joint pdf is initialized with the marginal cdf for each variable, and the whole process is recursively proceeded.

3.2.3 OBM-PCC method and its property

The traditional PCC allows to model the multivariate joint distribution from the marginal distribution by assuming all the variables are pair-wisely connected. However, in real situations, it is rare that all the nodes connect with each other [33]. For example, we model the M/M/1 queuing using the four-variate D-vine and the OBM-PCC, respectively. The results are shown in Fig. 3.4, where the four-variate D-vine is rearranged from Fig. 5.2 to better characterize the (conditional) copulas in the queuing system. It is clear that the OBM-PCC represents the variable interactions in the M/M/1 correctly, while the ordinary PCC (D-vine) gives two redundant pair-copulas (edge 12 and edge 34|12). Moreover, the calculation load under pair-wise connection (ordinary PCC) will be unacceptable when we have large amount of nodes. As a result, the variable connection/disconnection information should be incorporated to improve the traditional PCC method. Bauer et al. proposed a general method for incorporating the structure information of the direct acyclic graph (DAG) into the PCC [57]. However, it requires complex numerical integration in the pair-copula estimation, which impedes the efficiency of the algorithm. In the KPI analysis, the two stage OBM works as a special case of the DAG, where we can reasonably assume the system input variables are independent with each other. This offers extra information compared with regular DAGs and provides benefits for the PCC process. Thus, We propose the OBM-PCC method to remove redundant edges (pair-copula) among nodes by taking the OBM structure information into the PCC, at the same time retain the recursive property in pair-copula estimation to keep the algorithm efficiency.

The OBM structure provides the dependence information (edges E) for the pair-copula construction. In an OBM $G(V, E)$, for every node $i \in V$, we order the elements of $pa(i)$

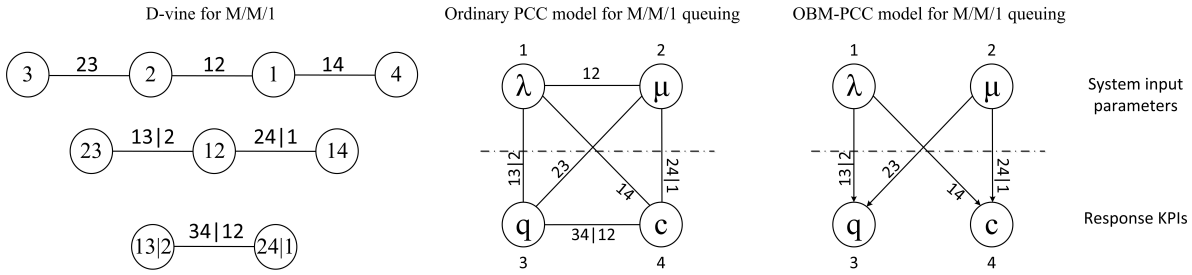


Figure 3.4: Difference and relationship between ordinary PCC and OBM-PCC.

increasingly and set:

$$pa(i;w) \triangleq \{\kappa \in pa(i) | \mathcal{S}_i(\kappa) < \mathcal{S}_i(w)\}, \quad w \in pa(i), \tag{3.12}$$

where $\mathcal{S}_i(\cdot)$ is the specified order for $pa(i)$. This order can be specified by the Kendall's tau [54], which is introduced in the appendix 3.5.1.

The example in the D-vine reveals that each edge in the OBM corresponds to a pair-copula, which can be specified based on Equation (5.14) as:

$$c_{iw|pa(i;w)}\Big(F_{i|pa(i;w)}(x_i|\mathbf{x}_{pa(i;w)}), F_{w|pa(i;w)}(x_w|\mathbf{x}_{pa(i;w)})\Big), \quad \forall i \in V, \forall w \in pa(i), \tag{3.13}$$

Under this situation, the number of (conditional) pair-copulas to be estimated reduces from $\binom{p}{2}$ to $\sum_{i=1}^{p} |pa(i)|$, where $|\cdot|$ represents the number of elements in a set. Moreover, the OBM-PCC method retains the recursive property when we construct all the pair-copulas. More specifically, we can have the lemma for the recursive property as follows:

Lemma 2: In a two-stage OBM with p variables, the p-dimension joint distribution can be recursively constructed by $\sum_{i=1}^{p} |pa(i)|$ (conditional) pair-copulas:

$c_{iw|pa(i;w)}\Big(F_{i|pa(i;w)}(x_i|\mathbf{x}_{pa(i;w)}),\ F_{w|pa(i;w)}(x_w|\mathbf{x}_{pa(i;w)})\Big),\ \forall i \in V, \forall w \in pa(i).$

The proof of the lemma is in the appendix 3.5.2.

The four nodes OBM in Fig. 3.2 can help to understand the basic idea of the lemma 2. In Fig. 3.2, if we assume $\mathcal{S}_3(2) < \mathcal{S}_3(1)$ and $\mathcal{S}_4(1) < \mathcal{S}_4(2)$, then $pa(1;\varnothing) = \varnothing, pa(2;\varnothing) = \varnothing, pa(3;1) = \{2\}, pa(4;1) = \varnothing, pa(3;2) = \varnothing$ and $pa(4;2) = \{1\}$. Consequently, the joint pdf in Fig. 3.2 contains only 4 (conditional) pair-copulas: $c_{23}\Big(F_2(x_2), F_3(x_3)\Big), c_{14}\Big(F_1(x_1), F_4(x_4)\Big)$, $c_{13|2}\Big(F_{1|2}(x_1|x_2), F_{3|2}(x_3|x_2)\Big)$ and $c_{24|1}\Big(F_{2|1}(x_2|x_1), F_{4|1}(x_4|x_1)\Big)$. Note that the four-variate D-vine in Section 3.2.2 requires 6 (conditional) pair copulas. In Fig. 3.2, $c_{23}\Big(F_2(x_2), F_3(x_3)\Big)$ and $c_{14}\Big(F_1(x_1), F_4(x_4)\Big)$ can be estimated from the marginal cdfs, which provide the conditional cdfs $F_{3|2}(x_3|x_2)$ and $F_{4|1}(x_4|x_1)$ for the rest two conditional pair-copulas as follows:

$$\begin{aligned} F_{3|2}(x_3|x_2) &= \frac{\partial C_{23}\Big(F_2(x_2), F_3(x_3)\Big)}{\partial F_2(x_2)} \\ F_{4|1}(x_4|x_1) &= \frac{\partial C_{14}\Big(F_1(x_1), F_4(x_4)\Big)}{\partial F_1(x_1)}, \end{aligned} \tag{3.14}$$

However, the conditional cdf $F_{2|1}(x_2|x_1)$ and $F_{1|2}(x_1|x_2)$ is not available. This is due to the OBM removes the edge between node 1 and 2 based on the OBM structure that the inputs for the queuing are independent. As a result, we also apply $X_1 \perp\!\!\!\perp X_2$ to the conditional cdf and find $F_{2|1}(x_2|x_1) = F_2(x_2)$, $F_{1|2}(x_1|x_2) = F_1(x_1)$. Under this situation, the conditional pair-copulas in Fig. 3.2 can be recursively specified from the marginal cdfs, which offers the computational efficiency.

The lemma 2 provides an efficient way for representing the joint distribution with pair-copulas when the structure information among variables is available. Furthermore, due to the recursive property in lemma 2, the OBM-PCC method guarantees the calculation efficiency.

3.2.4 Estimate the OBM-PCC model

The OBM-PCC model introduced in Section 3.2.3 decomposes the joint distribution into (conditional) pair-copulas. To correctly approximate the joint distribution, we need to identify the types of these pair-copulas and estimate the parameters in the corresponding ones. With reference to Sklar's theorem, we restrict our considerations to the joint distribution with uniform [0, 1] univariate margins. In real data application, empirical cdfs [67] can be applied to transform the data to uniform [0, 1]. More specifically, given n samples of a p-dimensional random vector $\mathbf{X}^k = (X_1^k, ..., X_p^k)$ with $k = 1, ..., n$, we construct the pseudo-observations $\mathbf{u}^k = (u_1^k, ..., u_p^k)$ by first retaining the rank r_i^k of the variable X_i among all variables $X_1, ..., X_p$ in the kth sample. Then, we can scale each element in $\mathbf{u}^k$ by a factor $n + 1$ to ensure that all values are inside $(0, 1)$:

$$\begin{aligned} u_i^k &= T(X_i^k) \\ &= \frac{r_i^k}{n+1} = \frac{\sum_{t=1}^{n} \mathbf{1}(X_i^t \leq X_i^k)}{n+1}, \end{aligned} \tag{3.15}$$

where $T(\cdot)$ is the transformation function for the original observation to scale as the pseudo-observations.

With these pseudo-observations, the OBM-PCC model yields the log-likelihood function as:

$$L(\boldsymbol{\theta}; \mathbf{u}) = \sum_{k=1}^{n} \sum_{i \in V} \sum_{w \in pa(i)} \log c_{iw|pa(i;w)} \Big(F_{i|pa(i;w)}(u_i^k | \mathbf{u}_{pa(i;w)}^k), F_{w|pa(i;w)}(u_w^k | \mathbf{u}_{pa(i;w)}^k); \boldsymbol{\theta} \Big), \tag{3.16}$$

where $\boldsymbol{\theta} = (\boldsymbol{\theta}_{iw|pa(i;w)})_{i \in V, w \in pa(i)}$ is a parameter vector for the pair-copula parameter of the corresponding edge in the OBM, $\mathbf{u} = (\mathbf{u}^1, ..., \mathbf{u}^n)$ with $\mathbf{u}^k \in [0, 1]^p, n \in \mathbb{N}, k = 1, \cdots, n$ is n i.i.d. samples of the random variables $\mathbf{U} = (u_1, ..., u_p)$. The order of the parent can

be specified based the Kendall's tau between parent nodes and their shared child node. Higher Kendall's tau between a parent and a specific child node will give higher priority to the parent in the order [68]. To correctly specify the MLE of Equation (5.19), we are concerned not only the parameter vector $\boldsymbol{\theta}$, but the specific type of the pair-copula for each edge in the OBM. To complete the comprehensive inference for the OBM-PCC model, we give algorithm 2 to specify the type of each copula and the initial value for the MLE of Equation (5.19), where we choose the Akaike's information criterion (AIC) [69] to select the pair-copula. The AIC in copula selection is comprehensively studied in [70], where a large scale simulation study shows that copula selection using the AIC is more reliable than that using goodness-of-fit tests (based on probability integral transform). In general, AIC can give stable result with quick response, which makes it a widely used criterion in selecting copula [64].

Algorithm 2 Identify the pair-copula and initial value of $\boldsymbol{\theta}$

Input: Transformed pseudo-observations $\mathbf{u}^1, ..., \mathbf{u}^n$.

Initialize Count variable C_v=0.

for $C_v = 0 \rightarrow |V|$ **do**

1. Find all pair-copulas with $|pa(i; w)| = C_v$;
2. For each such pair-copula, estimate the parameter(s) through MLE by assuming the copula type is Gaussian, Student, Frank, Clayton, Gumbel as well as the reflected Frank, reflected Clayton and reflected Gumbel;
3. Use AIC to select the best copula type and record the parameter value.

end for

Output: Each pair-copula's type and parameter values.

With the pair-copula's type and the corresponding parameter initial values from the algorithm 2, the Equation (5.19) can be numerically optimized. The candidate copulas in algorithm 2 are typical copulas utilized in the pair-copula modeling [71] and we use the AIC to choose the appropriate pair-copula for each edge. The details about these copulas can be found in the appendix 3.5.3 - 3.5.7.

To further estimate the response KPIs given specific system input parameter values, we can sample from the constructed OBM-PCC model to approximate the joint pdf $f(\mathbf{x})$. The sampling strategy is discussed in [64]. Here we give brief steps on approximating the $f(\mathbf{x})$.

First, sample $\omega_1, ..., \omega_p$ independently from the uniform distribution on [0,1]. Then, set samplings as:

$$u_i = \begin{cases} \omega_1, & i = 1 \\ F^{-1}_{i|1,...,i-1}(\omega_i|u_1, ..., u_{i-1}; \boldsymbol{\theta}), & i = 2, ..., p, \end{cases} \tag{3.17}$$

For example, the sampling for the OBM-PCC in Fig. 3.2 is as follows:

$$\begin{aligned} u_1 &= \omega_1, \\ u_2 &= F^{-1}_{2|1}(\omega_2|u_1) = F^{-1}_2(\omega_2) = \omega_2, \\ u_3 &= F^{-1}_{3|12}(\omega_3|u_1, u_2; \boldsymbol{\theta}) = h^{-1}_{32|1}(h_{31}(\omega_3, u_1; \boldsymbol{\theta}_{31}), u_2; \boldsymbol{\theta}_{32|1}) \\ u_4 &= F^{-1}_{4|123}(\omega_4|u_1, u_2, u_3; \boldsymbol{\theta}) = h^{-1}_{42|1}(h_{41}(\omega_4, u_1; \boldsymbol{\theta}_{41}), u_2; \boldsymbol{\theta}_{42|1}), \end{aligned} \tag{3.18}$$

where we use the function $h_{rv}(r, v, \Theta)$ to represent the conditional distribution function when r and v are uniform, i.e., $f(r) = f(v) = 1, F(r) = r$ and $F(v) = v$. That is:

$$h_{rv}(r, v, \Theta) = F_{r|v}(r|v) = \frac{\partial C_{rv}(r, v; \Theta)}{\partial v}, \tag{3.19}$$

where the second parameter of $h(\cdot)$ always corresponds to the conditioning variable and Θ denotes the set of parameters for the bi-variate copula. Further, let $h^{-1}_{rv}(r, v; \Theta)$ be the inverse of the h-function with respect to the first variable r, or equivalently the inverse of the conditional distribution function. The details about the candidate h and h^{-1} functions

are in the appendix 3.5.3 - 3.5.7.

After obtaining the PCC samples, we can use the inverse empirical cdf $T^{-1}(\mathbf{u}^k)$ to get the samples for $(x_1^k, ...x_p^k)$, which results in the numerical joint pdf $\hat{f}(\mathbf{x})$. The numerical conditional distribution by conditioning on system input parameters can also be obtained. For example, we can get $\hat{f}(q, c|\lambda, \mu)$, $\hat{f}(q|\lambda, \mu)$ and $\hat{f}(c|\lambda, \mu)$ for the OBM-PCC model in Fig. 3.2. The performance of the proposed OBM-PCC model in estimating response KPIs will be introduced in Section 3.3.

3.2.5 Summary of the OBM-PCC method

Section 3.2.2 to Section 3.2.4 introduce the OBM-PCC model and the joint distribution approximation method for system input parameters and response KPIs. We give a comprehensive summary to numerically approximate the joint pdf of KPIs, which is shown in the following algorithm 3.

Algorithm 3 Numerical approximation of $f(x_1, \cdots, x_p)$

Input: Sample vectors for each KPI variables $\mathbf{x}_1, ..., \mathbf{x}_p$;
The system input parameter and response KPI variables sets $\mathbf{I}$ and $\mathbf{O}$;
The size of system input parameter variables $l = |\mathbf{I}|$.

1. Identify OBM structure based on the physical information or statistical tests [37];
2. Decompose the pair-copula based on the OBM structure to get the OBM-PCC model;
3. Employ Equation (5.18) to get the pseudo-observations $\mathbf{u}^1, ..., \mathbf{u}^n$;
4. Use algorithm 1 and MLE to estimate the OBM-PCC model parameters;
5. With the OBM-PCC model and Equation (5.20), sample in the pseudo-observation space;
6. Take the inverse of Equation (5.18) to the pseudo-observation samples to obtain KPI samples;
7. Numerically approximate $f(\mathbf{x})$ with the KPI samples.

Output: $\hat{f}(\mathbf{x})$.

3.3 Numerical studies

In this section, extensive numerical studies are conducted to demonstrate the effectiveness of the proposed OBM-PCC method. More specifically, we will apply our method to the M/M/1 queue, G/G/1 queue and a serial production system with 4 servers to approximate the joint distribution of system input parameters and response KPIs in these systems. Based on the approximated joint distribution, we will demonstrate the conditional distribution for each response KPI given the system input parameters. To validate the approximated conditional distributions, we will perform the Kolmogorov-Smirnov test (K-S test) to evaluate the goodness-of-fit of the approximated distribution to the underlying true distribution [72, 73]. The K-S test is a non-parametric test of the equality of continuous probability distributions that can be used to compare a sample with a theoretic probability distribution (one-sample K-S test), or to compare two samples (two-sample K-S test). The one-sample K-S test evaluates the goodness-of-fit between the observed distribution function (ODF) and the theoretical distribution (TDF). The two-sample K-S test is used to test whether two ODFs differ with each other. We define D_1 as the largest discrepancy between the ODF and TDF, and D_2 as the largest discrepancy between two ODFs. More technical details about the one-sample K-S test and two-sample K-S test is in the appendix 3.5.8. The K-S test is chosen to validate our method's performance due to the following advantages: (i) It is easy to compute and the critical values are tabulated. (ii) The statistical properties are distribution-free. (iii) It is always consistent against all alternative continuous distribution functions, i.e., the power of the tests approaches 1 as the sample size goes to infinity.

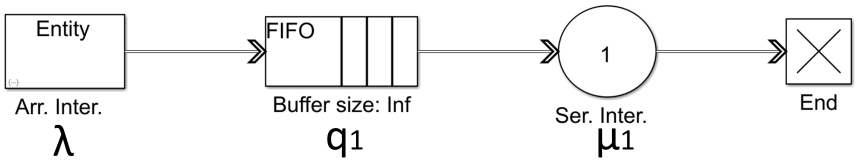
3.3.1 M/M/1 system

First, we consider the M/M/1 queuing system with the first come first serve (FCFS) dispatching rule for waiting lines. The diagram for M/M/1 queue is shown in Fig. 5.5 (a),

where we choose infinity buffer size to demonstrate our method. The simulation parameters are listed in Table 3.1. In the simulation, we first randomly pick a pair of λ and μ_1 from the corresponding uniform distributions. Then, the arrival and service distributions are set according to Table 3.2 for the M/M/1 queue. As the simulation goes to the steady state, we can sample and record q_1 and c. Then, we re-pick another pair of λ and μ_1 and repeat the process. The simulation is implemented in Simulink, Matlab 2017. In our simulations, we sample 800 different pairs of system input parameters and the corresponding response KPIs as inputs for the algorithm 3. By implementing algorithm 3, we can obtain the OBM-PCC model, which is shown in Fig. 5.5 (b). The pair-copulas attached on the lines are the MLE results by algorithm 2. More detailed results are shown in Table 3.2, where we demonstrate the estimation results for 10 different pairs of system input parameters by giving mean and variance of the corresponding conditional distributions of response KPIs. The conditional distributions of different response KPIs are obtained from 1000 samples of the approximated joint distribution $\hat{f}(\mathbf{x})$, which is the output of algorithm 3. It is well known that steady state M/M/1 queue has analytical distributions for response KPIs. All the mean and variance values in Table 3.2 are validated with theoretical results, e.g., $E(c) = \frac{1}{\mu-\lambda}, Var(c) = \frac{1}{(\mu-\lambda)^2}$, and have relative error less than 5%. Besides comparisons in moment level, we also perform the one-sample K-S test to compare our model result with the theoretical truth in the distribution level. The 1000 samples for response KPIs are used to construct the ODF, which is compared with the analytical distribution TDF under the one-sample K-S test. We show the D_1 (the largest discrepancy between ODF and TDF) in Table 3.2, where we also give critical value S_1^D at 95% confidence level. It can be seen that all D_1s are smaller than the critical value S_1^D, which means we cannot reject the null hypothesis that the 1000 samplings are from the underlying true conditional distributions. Besides, D_1s in Table 3.2 maintain at small values indicate the approximated conditional

distributions are very close to those underlying truth. To have an overall view of the OBM-PCC results, we give 7000 samples of the mean of the conditional response KPIs reproduced by the 800 original simulation samples. Fig. 3.6 (a) and Fig. 3.7 (a) show the approximated samples track the simulation samples well and fill the gaps the simulation samples cannot reach. Fig. 3.6 (b) and Fig. 3.7 (b) are the comparisons between the approximated samples and the mean response surface from theoretical results, where we can see the approximated samples represent the theoretical response surface very well.

(a) M/M/1 queue diagram.



(b) OBM-PCC modeling result.

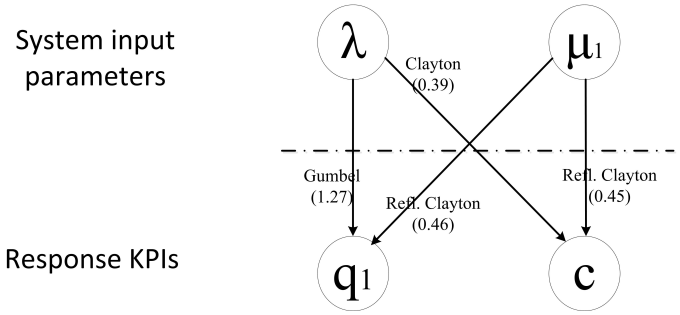


Figure 3.5: Simulation for M/M/1 queue.

Table 3.1: Parameter setting for M/M/1 queue.

Parameter	Type	Mean	Variance	Min	Max
Arrival interval	Exponential	$\frac{1}{\lambda}$	$\frac{1}{\lambda^2}$	0	∞
Service interval	Exponential	$\frac{1}{\mu_1}$	$\frac{1}{\mu_1^2}$	0	∞
λ	Uniform	3.5	$\frac{1}{12}$	3	4
μ_1	Uniform	5	$\frac{1}{12}$	4.5	5.5

(a) Comparison between original samples and approximated samples. (b) Comparison between approximated samples and theoretical response surface.

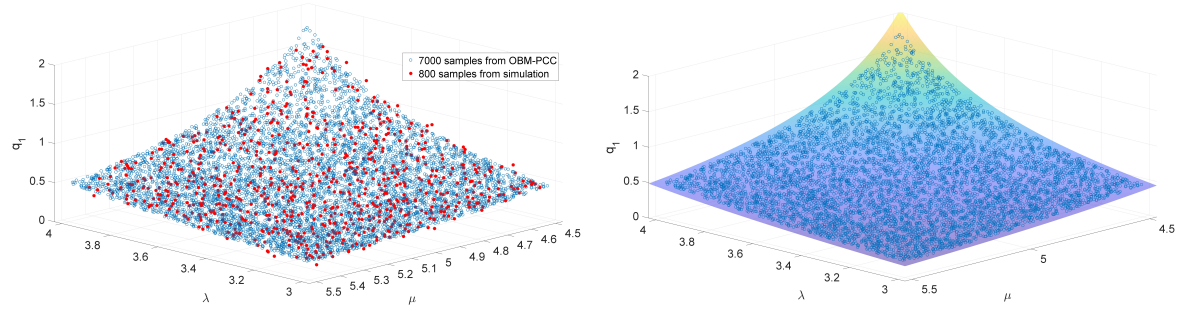


Figure 3.6: Comparison of waiting time in M/M/1 queue.

(a) Comparison between original samples and approximated samples. (b) Comparison between approximated samples and theoretical response surface.

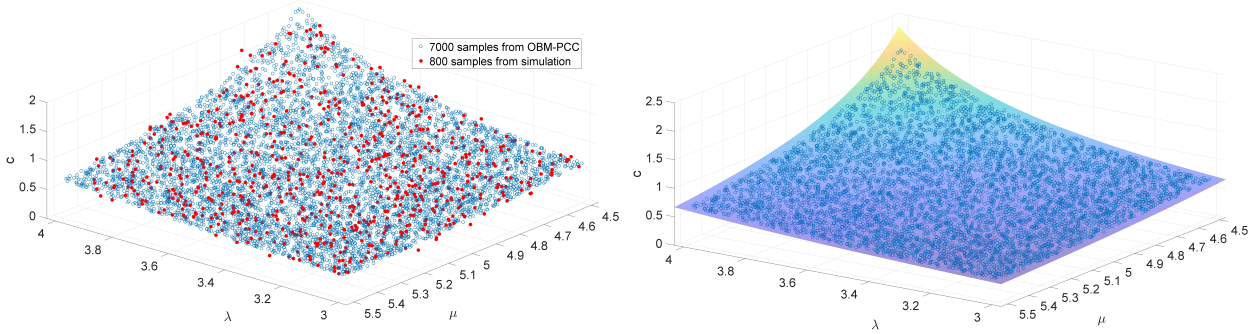


Figure 3.7: Comparison of cycle time in M/M/1 queue.

Table 3.2: Detailed results for M/M/1 queue.

Input parameter		λ	3.05	3.85	3.25	3.48	3.32	3.17	3.01	3.80	3.89	3.12
		μ_1	5.24	4.96	4.99	4.72	5.13	5.26	4.71	5.49	5.31	5.38
Response KPIs (conditional)	q_1	Sample Mean	0.268	0.709	0.383	0.610	0.369	0.290	0.379	0.413	0.520	0.278
		Sample variance	0.174	0.829	0.311	0.674	0.292	0.198	0.321	0.344	0.499	0.184
		$D_1(S_1^D = 0.043)$	0.021	0.017	0.010	0.026	0.014	0.011	0.012	0.006	0.018	0.007
	c	Sample Mean	0.456	0.906	0.578	0.815	0.560	0.476	0.589	0.591	0.708	0.458
		Sample variance	0.208	0.844	0.348	0.701	0.328	0.231	0.359	0.362	0.528	0.216
		$D_1(S_1^D = 0.043)$	0.030	0.023	0.014	0.034	0.018	0.017	0.021	0.007	0.027	0.021

3.3.2 G/G/1 system

Generally, the G/G/1 queuing model provides more flexibility in approximating real systems compared to M/M/1 queues. However, the stationary distribution of response KPIs in G/G/1 queue is hard to derive analytically. Therefore, we compare the performance of our method in G/G/1 queuing with simulated moments and distributions. The parameter sets are shown in Table 3.3 and we choose infinity buffer size to demonstrate our method. Similar to that in Section 3.3.1, we have 800 samples of different pairs of system input parameters and the corresponding response KPIs as training data. Fig. 3.8 shows the OBM-PCC results. Table 3.4 gives details about 10 different sets of system input parameters and the corresponding response KPIs distribution information. Different with that in Section 3.3.1, all the mean and variance in Table 3.3 are compared with simulated mean and variance from G/G/1 simulation rather than theoretical results. Here we treat the results from the G/G/1 simulation as underlying truth. Relative error less than 5% for these mean and variance are validated. We also perform the two-sample K-S test for the distributions, where we sample 1000 times from our approximated distribution and 1000 times from the G/G/1 simulation to construct two ODFs. We report the D_2 and the critical value S_2^D at 95% confidence level in Table 3.3. It can be seen that all the D_2s are smaller than the critical value S_2^D, which means we cannot reject the null hypothesis that the 1000 samplings are from the same distribution. Besides, we also give the overall view of the between original samples and the approximated samples in Fig. 3.9. We can see the approximated samples well match the original samples, which verifies the effectiveness of our method in modeling G/G/1 queuing.

Table 3.3: Parameter setting for G/G/1 queue.

Parameter	Type	Mean	Variance	Min	Max
Arrival interval	Log-normal	$\frac{1}{\lambda}$	1	0	∞
Service interval	Erlang	$\frac{1}{\mu_1}$	$\frac{1}{2\mu_1^2}$	0	∞
λ	Uniform	3.5	$\frac{1}{12}$	3	4
μ_1	Uniform	5.5	$\frac{1}{12}$	5	6

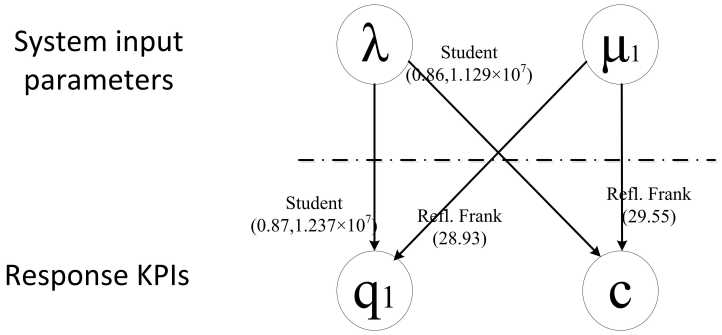


Figure 3.8: Simulation result for G/G/1 queue.

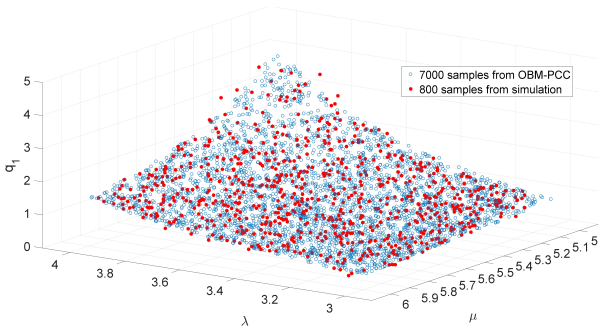
Table 3.4: Detailed results for G/G/1 queue.

Input parameter		λ	3.13	3.28	3.85	3.10	3.19	3.15	3.23	3.20	3.51	3.71
		μ_1	5.91	5.55	5.93	5.82	5.49	5.26	5.91	5.74	5.88	5.99
Response KPIs (conditional)	q_1	Sample Mean	0.536	0.805	1.242	0.542	0.737	0.825	0.596	0.632	0.854	0.996
		Sample variance	0.404	0.804	1.232	0.543	0.741	0.832	0.599	0.636	0.853	0.987
		$D_2(S_2^D = 0.043)$	0.040	0.033	0.034	0.041	0.028	0.027	0.031	0.017	0.037	0.031
	c	Sample Mean	0.705	0.986	1.409	0.713	0.919	1.016	0.765	0.806	1.023	1.161
		Sample variance	0.418	0.814	1.651	0.431	0.703	0.863	0.490	0.544	0.885	1.131
		$D_2(S_2^D = 0.043)$	0.031	0.027	0.029	0.036	0.024	0.021	0.022	0.010	0.028	0.017

3.3.3 Serial production lines

In this section, we consider a serial production system consisting of four workstations (servers). The diagram of the production line is shown in Fig. 3.10 (a) and the parameters

(a) Original samples and OBM-PCC samples for waiting time.



(b) Original samples and OBM-PCC samples for cycle time.

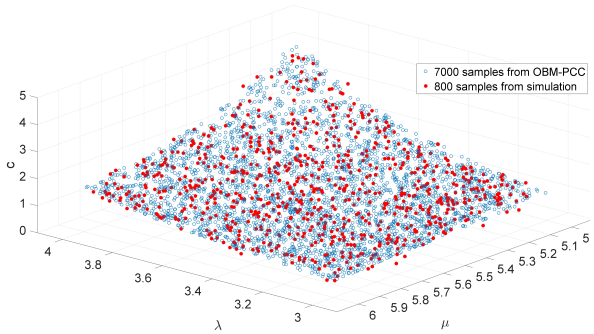


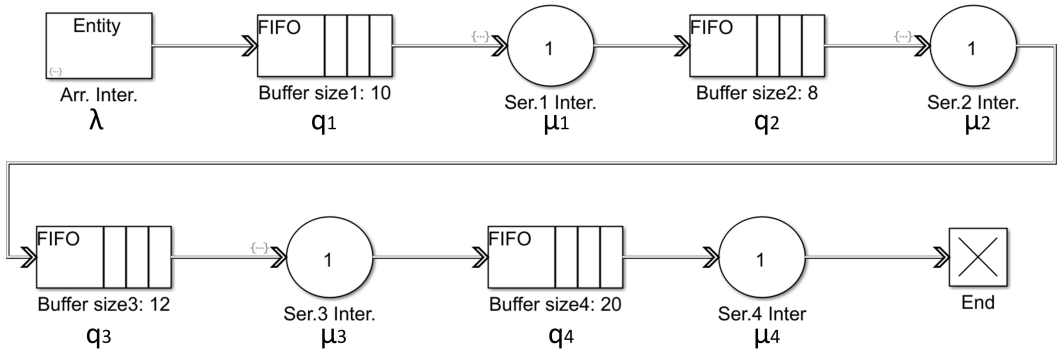
Figure 3.9: Comparisons of waiting time and cycle time for G/G/1 queue.

are in Table 3.5. In this case, we use finite buffer size between servers, and the buffer size for each queue is shown in Fig. 3.10 (a). Similar steps as those in Section 3.3.1 and 3.3.2 are followed, which result in the OBM-PCC for the production line in Fig. 3.10 (b).

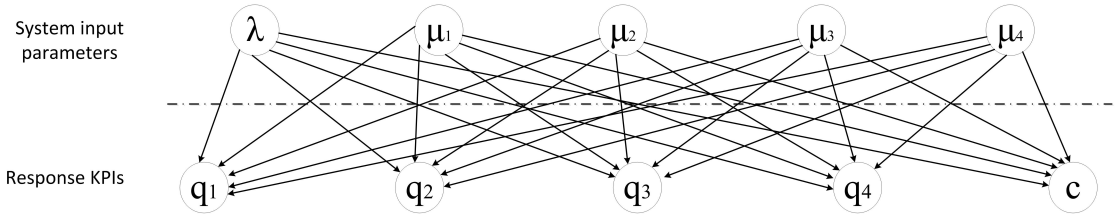
Detailed analysis for each response KPIs are shown in Table 3.6, where we show 10 different sets of system input parameters and the corresponding response KPIs features. Again, the serial G/G/1 simulations are conducted and act as the underlying truth to compare with our method's performance. All the mean and variance values in Table 3.6 are validated, and have relative error less than 8%. We also perform the two-sample K-S test for the distributions and report the D_2 and the critical value S_2^D at 95% confidence level. It can be seen that all the D_2s are smaller than the critical value S_2^D, which means we cannot reject the null hypothesis that the 1000 samplings are from the same distribution. Due to the high-dimension of system input parameters, we are unable to visually show the comparison between original samples and the approximated samples. Nevertheless, the two-sample K-S test verifies the effectiveness of our methods in the serial production lines.

To illustrate the computation load of the proposed method, we give a comparison of computational time in Table 3.7. Specifically, after we obtain the OBM-PCC model, we randomly generate 7000 samples of the system input parameter and the corresponding

(a) Serial production line with 4 G/G/1 queues.



(b) OBM-PCC modeling result.



λ→q1: Student (0.74, 1.011×10^7); λ→q2: Student (0.71, 1.000×10^7); λ→q3: Student (0.68, 9.774×10^6); λ→q4: Student (0.68, 9.608×10^6); λ→c: Student (0.67, 9.559×10^6)
μ1→q1: Refl. Frank (23.93); μ1→q2: Refl. Frank (15.22); μ1→q3: Refl. Frank (5.12); μ1→q4: Refl. Frank (2.06); μ1→c: Refl. Frank (14.47)
μ2→q1: Refl. Frank (16.09); μ2→q2: Refl. Frank (25.52); μ2→q3: Refl. Frank (14.33); μ2→q4: Refl. Frank (6.80); μ2→c: Refl. Frank (15.19)
μ3→q1: Refl. Frank (4.92); μ3→q2: Refl. Frank (14.11); μ3→q3: Refl. Frank (22.28); μ3→q4: Refl. Frank (13.76); μ3→c: Refl. Frank (13.12)
μ4→q1: Refl. Frank (1.88); μ4→q2: Refl. Frank (3.48); μ4→q3: Refl. Frank (12.89); μ4→q4: Refl. Frank (20.06); μ4→c: Refl. Frank (11.19)

Figure 3.10: Simulation result for serial production line.

response KPIs (sample from the OBM-PCC based joint distribution), and record the computational time. For the same 7000 system input parameters, we run the queuing simulation to obtain the corresponding response KPIs, and record the computational time. We repeat these procedures 100 times and list the mean value of time consumption.

From Table 3.7, we can see that the OBM-PCC model needs significantly less time compared with the simulation model. This is because the simulation model needs to reach steady state for each sample, which is quite time-consuming. All the calculation is implemented in Simulink, Matlab 2017.

Table 3.5: Parameter setting for the serial production line.

Parameter	Type	Mean	Variance	Min	Max
Arrival interval	Log-normal	$\frac{1}{\lambda}$	1	0	∞
Service 1 interval	Erlang	$\frac{1}{\mu_1}$	$\frac{1}{2\mu_1^2}$	0	∞
Service 2 interval	Erlang	$\frac{1}{\mu_2}$	$\frac{1}{2\mu_2^2}$	0	∞
Service 3 interval	Erlang	$\frac{1}{\mu_3}$	$\frac{1}{2\mu_3^2}$	0	∞
Service 4 interval	Erlang	$\frac{1}{\mu_4}$	$\frac{1}{2\mu_4^2}$	0	∞
λ	Uniform	2	$\frac{1}{12}$	1.5	2.5
μ_1	Uniform	5	$\frac{1}{3}$	4	6
μ_2	Uniform	4.5	$\frac{1}{3}$	3.5	5.5
μ_3	Uniform	4	$\frac{1}{3}$	3	5
μ_4	Uniform	3.5	$\frac{1}{12}$	3	4

From the extensive case studies, we can find that the proposed OBM-PCC method can be successfully applied to a wide class of production systems with satisfactory accuracy. Note that although the traditional simulation method can mimic the underlying truth, it is time consuming to set up the simulation for different system input parameters due to 'warm-up' period in the simulation. On the other hand, our method can approximate response KPIs under any configuration of system input parameters once we complete the modeling, which is efficient and desirable in practice use. This property is particularly useful when we are interested in specific systems input parameters' influence on the response KPIs, where we can numerically marginalize the miscellaneous input parameters and focus on specific input parameters. For example, we can get $\hat{f}(q_1, q_2|\lambda)$ in the serial production line to investigate the influence on the queuing time only from the arrival rate. This is difficult to evaluate through simulations since the simulation cannot deal with the marginalization on specific parameters.

Table 3.6: Detailed results for serial production line.

Input parameter		λ	1.64	2.16	1.94	1.85	2.10	2.44	1.59	1.71	2.12	1.67
		μ_1	4.84	4.07	4.76	5.66	4.53	5.75	5.63	4.60	5.36	5.96
		μ_2	5.33	5.20	5.03	4.67	4.81	4.60	5.14	4.44	4.29	4.93
		μ_3	4.58	4.87	4.59	4.10	4.39	4.25	4.44	3.46	3.73	4.00
		μ_4	3.96	3.68	3.19	3.92	3.75	3.59	3.15	3.84	3.99	3.47
Response KPIs (conditional)	q_1	Sample Mean	0.119	0.461	0.209	0.118	0.303	0.263	0.049	0.156	0.198	0.075
		Sample variance	0.051	0.348	0.094	0.039	0.176	0.140	0.009	0.062	0.089	0.024
		$D_2(S_2^D = 0.062)$	0.025	0.031	0.028	0.021	0.027	0.018	0.024	0.038	0.040	0.035
	q_2	Sample Mean	0.072	0.123	0.136	0.172	0.177	0.367	0.086	0.124	0.311	0.103
		Sample variance	0.017	0.043	0.049	0.071	0.074	0.215	0.017	0.036	0.174	0.028
		$D_2(S_2^D = 0.062)$	0.045	0.038	0.038	0.027	0.034	0.021	0.037	0.048	0.024	0.029
	q_3	Sample Mean	0.147	0.148	0.157	0.211	0.221	0.382	0.218	0.283	0.417	0.191
		Sample variance	0.059	0.064	0.065	0.099	0.120	0.276	0.093	0.159	0.320	0.085
		$D_2(S_2^D = 0.062)$	0.032	0.019	0.041	0.027	0.030	0.028	0.032	0.043	0.032	0.038
	q_4	Sample Mean	0.152	0.395	0.545	0.202	0.339	0.784	0.182	0.166	0.237	0.223
		Sample variance	0.063	0.306	0.628	0.109	0.233	1.013	0.084	0.083	0.129	0.120
		$D_2(S_2^D = 0.062)$	0.039	0.026	0.042	0.030	0.029	0.023	0.042	0.033	0.026	0.026
	c	Sample Mean	1.287	2.061	1.990	1.583	1.977	2.727	1.286	1.667	2.115	1.454
		Sample variance	0.254	1.072	1.163	0.483	0.950	2.396	0.297	0.508	1.109	0.357
		$D_2(S_2^D = 0.062)$	0.051	0.048	0.055	0.041	0.058	0.038	0.050	0.054	0.049	0.047

Table 3.7: Time consumption comparison for generating 7000 samples.

Models	OBM-PCC			Simulation model		
Queuing system	M/M/1	G/G/1	4 serial G/G/1	M/M/1	G/G/1	4 serial G/G/1
Time consumption	7.781s	8.111s	51.346s	29112s	29774s	86956s

To guarantee the model performance, we give several important guidelines for simulation data generation and prediction procedures:

1 We suggest sample system input parameters from uniform distributions. This procedure provides convenience for copula estimation that needs standard uniform data.

2 The simulated queuing system should be able to reach the steady state, e.g., $\lambda < \mu$.

3 The response KPIs should be sampled only after the system reaches steady state. This can be guaranteed by checking the steady state of the Markov Chains [74, 75].

4 The support of uniform distribution of system input parameters should cover the prediction scope to enable the interpolation rather than the extrapolation for the prediction. This is because the copula based extrapolation has the extreme value problems [76, 77]

5 The larger the sample size of simulation data, the more accurate the estimation result. We suggest take at least 800 samples according to the Dvoretzky–Kiefer–Wolfowitz inequality [78].

3.4 Conclusion

In this chapter, we propose to approximate the KPIs joint distribution through sampling from the OBM-PCC model. First, the KPIs are modeled as random variables and the process is formulated under the OBM framework. Then, the pair-copulas are constructed based on the OBM structure to approximate the KPIs joint distribution. The feature of the proposed method is that it focuses on the distribution level approximation rather than the individual KPI estimation, which offers the feasibility in high dimensional modeling for KPIs under any configurations. Moreover, compared with traditional simulation method, the proposed method is more efficient due to the closed-form sampling procedures. The effectiveness and accuracy of the proposed method are verified through various numerical

studies.

Some interesting open issues remain in the proposed method. First, the approximated distribution can be further summarized into monitoring statistics to conduct KPI condition monitoring and prediction. The KPI monitoring based on the proposed method benefits from not only the rich information embedded in the approximated joint distribution, but the elimination of redundant relationships due to OBM structure. These features provide extra information for the KPI monitoring and potentially offer quicker response to change detection. Second, the domain knowledge incorporation in the OBM-PCC can be further studied. In our current work, the domain knowledge mainly helps build the OBM and reduce the PCC calculation load. A more efficient way to employ domain knowledge is to directly benefit the PCC process. One possible way is to extract PCC domain knowledge, e.g, copula type and parameter, from existing processes, then transfer such knowledge to the PCC for target process. However, the variable inconsistency in different processes sets difficulties in fusing and transferring information in this approach. Another possible topic based on this work is to do the KPI control and optimization for system input parameters. For example, given a specific required waiting time distribution, we want to find out the optimal arrival and service rate parameters or distributions. Since we can approximate the KPIs joint distribution, a simple way to conduct this KPI control task is brute-force search over the system input parameter space. However, the high-dimension of the system input parameter space poses difficulties in the search procedure. We will study these topics in future work.

3.5 Appendix

3.5.1 Kendall's tau

The Kendall's tau is widely used as the dependence measure in the copula analysis due to its scale-invariant property [54]. The exact relationship between the Kendall's tau and various copula parameters is shown in Table 3.8, where the larger the Kendall's tau, the stronger the dependence.

Table 3.8: Kendall's tau of various bi-variate copulas.

Copula	Gaussian	Student	Clayton	Gumbel	Frank
Parameter	ρ	ρ, ν	δ	δ	δ
Kendall's τ	$\frac{2}{\pi}\arcsin(\rho)$	$\frac{2}{\pi}\arcsin(\rho)$	$\frac{\delta}{\delta+2}$	$\frac{\delta-1}{\delta}$	$1+\frac{4\Big(D(\delta)-1\Big)}{\delta}$

$D(\delta) = \frac{1}{\delta}\int_0^{\delta} \frac{t}{\exp(t)-1}dt$

3.5.2 Proof of lemma 2

We first show that the p-dimension joint distribution can be represented by $\sum_{i=1}^{p}|pa(i)|$ (conditional) pair-copulas:

According to the OBM structure, $\{X_i, i \in \mathbf{I}\}$ are in the first stage and independent with each other. Thus, $|pa(i)| = 0, i \in \mathbf{I}$. The number of pair-copulas now only depends on the parent set of $\{X_i, i \in \mathbf{O}\}$, which is $\sum|pa(i)|, i \in \mathbf{O}$. Since $|pa(i)| = 0, i \in \mathbf{I}$, we have $\sum|pa(i)|, i \in \mathbf{O}$ equals to $\sum_{i=1}^{p}|pa(i)|$. Thus, the p-dimension joint distribution can be represented by $\sum_{i=1}^{p}|pa(i)|$ (conditional) pair-copulas.

Then, we will show that all the (conditional) pair-copulas can be recursively constructed from the marginal cdfs $F_1(x_1), \cdots, F_p(x_p)$. We give the proof through induction, where we will use the simplified notation, e.g., $c_{iw|pa(i;w)}$, to represent the (conditional) pair-copula.

$\mathcal{S}_i(w) = 1$:

The $c_{iw|pa(i;w)}$ is a unconditional pair-copula, it is easy to see the lemma holds.

$\mathcal{S}_i(w) = k$:

We suppose the lemma holds, which means all the $c_{iw|pa(i;w)}, \forall i \in V, \forall w \in pa(i)$ are available from $F_1(x_1), \cdots, F_p(x_p)$.

$\mathcal{S}_i(w) = k + 1$:

$c_{iw|pa(i;w)}$ depends on $F_{i|pa(i;w)}(x_i|\mathbf{x}_{pa(i;w)})$ and $F_{w|pa(i;w)}(x_w|\mathbf{x}_{pa(i;w)})$.

Note that $w \perp\!\!\!\perp pa(i;w)$, which results in $F_{w|pa(i;w)}(x_w|\mathbf{x}_{pa(i;w)}) = F_w(x_w)$.

According to Equation (5.13), we have $F_{i|pa(i;w)}(x_i|\mathbf{x}_{pa(i;w)}) = \frac{\partial C_{iw'|(pa(i;w)\setminus\{w'\})}}{\partial F_{w'|(pa(i;w)\setminus\{w'\})}}$, $\mathcal{S}_i(w') = k$.

Since $C_{iw'|(pa(i;w)\setminus\{w'\})}, \mathcal{S}_i(w') = k$ is available from $F_1(x_1), \cdots, F_p(x_p)$, $F_{i|pa(i;w)}(x_i|\mathbf{x}_{pa(i;w)})$ is also available.

Thus, $c_{iw|pa(i;w)}, \forall i \in V, \forall w \in pa(i)$ are available from $F_1(x_1), \cdots, F_p(x_p)$ when $\mathcal{S}_i(w) = k + 1$.

The lemma holds.

3.5.3 The bi-variate Gaussian copula

The density of the bi-variate Gaussian copula is given by

$$c(u, v; \rho) = \frac{1}{\sqrt{1-\rho^2}} \exp\left\{-\frac{\rho^2(x_1^2 + x_2^2) - 2\rho x_1 x_2}{2(1-\rho^2)}\right\}, \tag{3.20}$$

where $\rho \in (-1, 1)$ is the parameter of the copula, $x_1 = \Phi^{-1}(u)$, $x_2 = \Phi^{-1}(v)$ and $\Phi^{-1}(\cdot)$ is the inverse of the standard univariate Gaussian distribution function. For this copula, the

h-function is given by

$$h_{uv}(u, v; \rho) = \Phi\left(\frac{\Phi^{-1}(u) - \rho\Phi^{-1}(v)}{\sqrt{1-\rho^2}}\right), \tag{3.21}$$

and the inverse of the h-function is :

$$h_{uv}^{-1}(u, v; \rho) = \Phi\Big(\Phi^{-1}(u)\sqrt{1-\rho^2} + \rho\Phi^{-1}(v)\Big), \tag{3.22}$$

The dependence represented by a bi-variate Gaussian copula is $\frac{2}{\pi}\arcsin(\rho)$.

3.5.4 The bi-variate Student copula

The density of the bi-variate Student copula is given by

$$c(u, v; \rho, \tau) = \frac{\Gamma(\frac{\tau+2}{2})/\Gamma(\frac{\tau}{2})}{\tau\pi \mathrm{d}t(x_1, \tau)\mathrm{d}t(x_2, \tau)\sqrt{1-\rho^2}}\left\{1 + \frac{x_1^2 + x_2^2 - 2\rho x_1 x_2}{\tau(1-\rho^2)}\right\}^{-\frac{\tau+1}{2}}, \tag{3.23}$$

where $\rho \in (-1, 1)$ and $\tau \in \mathbb{N}$ are the parameters of the copula, $x_1 = t_\tau^{-1}(u)$, $x_2 = t_\tau^{-1}(v)$, and $\mathrm{d}t(\cdot, \tau)$ and $t_\tau^{-1}(\cdot)$ are the probability density and the quantile function, respectively, for the univariate standard Student-distribution with τ degrees of freedom. For this copula, the h-function is given by

$$h_{uv}(u, v; \rho, \tau) = t_{\tau+1}\left\{\frac{t_\tau^{-1}(u) - \rho t_\tau^{-1}(v)}{\sqrt{\frac{\tau + (t_\tau^{-1}(v))^2(1-\rho^2)}{\tau+1}}}\right\}, \tag{3.24}$$

and the inverse of the h-function is given by

$$h_{uv}^{-1}(u, v; \rho, \tau) = t_\tau\left\{t_{\tau+1}^{-1}(u)\sqrt{\frac{\tau + (t_\tau^{-1}(v))^2(1-\rho^2)}{\tau+1}} + \rho t_\tau^{-1}(v)\right\}, \tag{3.25}$$

The dependence represented by a bi-variate student copula is $\frac{2}{\pi}\arcsin(\rho)$.

3.5.5 The bi-variate Clayton copula

The density of the bi-variate Clayton copula is given by

$$c(u, v; \delta) = (1 + \delta)(u \cdot v)^{-1-\delta}(u^{-\delta} + v^{-\delta} - 1)^{-1/\delta-2}, \tag{3.26}$$

where $\delta \in (0, \infty)$ is the parameter of the copula. For this copula, the h-function is given by

$$h_{uv}(u, v; \delta) = v^{-\delta-1}(u^{-\delta} + v^{-\delta} - 1)^{-1-1/\delta}, \tag{3.27}$$

and the inverse of the h-function is given by

$$h_{uv}^{-1}(u, v; \delta) = \left((u \cdot v^{\delta+1})^{-\frac{\delta}{\delta+1}} + 1 - v^{-\delta}\right)^{-1/\delta}, \tag{3.28}$$

The dependence represented by a bi-variate Clayton copula is $\frac{\delta}{\delta+2}$.

3.5.6 The bi-variate Gumbel copula

Let $\tilde{u} = -\log u$ and $\tilde{v} = -\log v$. The density of the bi-variate Gumbel copula is given by

$$\begin{aligned} c(u, v; \delta) &= C(u, v; \delta)(u \cdot v)^{-1} \frac{(\tilde{u}\tilde{v})^{\delta-1}}{(\tilde{u}^{\delta} + \tilde{v}^{\delta})^{2-1/\delta}} \\ C(u, v; \delta) &= \exp\{-(\tilde{u}^{\delta} + \tilde{v}^{\delta})^{1/\delta}\}, \end{aligned} \tag{3.29}$$

where $\delta \in [1, \infty)$ is the parameter of the copula. For this copula, the h-function is given by

$$h_{uv}^{-1}(u, v; \delta) = v^{-1} \exp\Big(- (\tilde{u}^{\delta} + \tilde{v}^{\delta})^{1/\delta}\Big) \cdot \left(1 + \left(\frac{\tilde{u}}{\tilde{v}}\right)^{\delta}\right)^{-1+1/\delta}, \tag{3.30}$$

and the inverse of the h function must be obtained numerically using, for example, Newton-Raphson method [79]. The dependence represented by a bi-variate Gumbel copula is $1 - \frac{1}{\delta}$.

3.5.7 The bi-variate Frank copula

The density of the bi-variate Frank copula is given by

$$\begin{aligned} c(u, v; \delta) =& \frac{\delta\eta \exp\Big(- \delta(u + v)\Big)}{\Big(\eta - \Big(1 - \exp(\delta u)\Big)\Big(1 - \exp(\delta v)\Big)\Big)^2} \\ & \eta = 1 - \exp(-\delta), \end{aligned} \tag{3.31}$$

where $\delta \in [0, \infty)$ is the parameter of the copula. For this copula, the h-function is given by

$$h_{uv}^{-1}(u, v; \delta) = \frac{e^{-\delta v}}{\frac{1-e^{-\delta}}{1-e^{-\delta u}} + e^{-\delta v} - 1}, \tag{3.32}$$

and the inverse of the h-function is given by

$$h_{uv}^{-1}(u, v; \delta) = -\log\left\{1 - \frac{1 - e^{-\delta}}{(u^{-1} - 1)e^{\delta v} + 1}\right\} \Big/ \delta, \tag{3.33}$$

The dependence represented by a bi-variate Frank copula is $1 + \frac{4\Big(D(\delta) - 1\Big)}{\delta}$, where $D(\delta) = \frac{1}{\delta}\int_0^{\delta} \frac{t}{\exp(t)-1} dt$.

3.5.8 K-S test

Suppose we have samples $x_1, \cdots, x_n$ for a random variable, and a theoretical distribution function (TDF) $F(x)$. The one-sample K-S test shows:

$$\begin{aligned} &H_0 : x_1, \cdots, x_n \text{ are from } F(x). \\ &H_A : x_1, \cdots, x_n \text{ are not from } F(x). \\ &D_1 = \sup_d |\hat{F}_n(d) - F(d)| \\ &\text{Test statistic} : S_1 = \sqrt{n} \cdot D_1, \end{aligned} \tag{3.34}$$

where $\hat{F}_n(d)$ is the observed distribution function (ODF) from the samples. It is clear that D_1 is the the largest vertical discrepancy between ODF and TDF, as shown in Fig. 5.4. The one-sample K-S test gives a measure of the distance between the empirical distribution function of samples and $F(x)$. Similarly, we can have the two-sample K-S test for samples

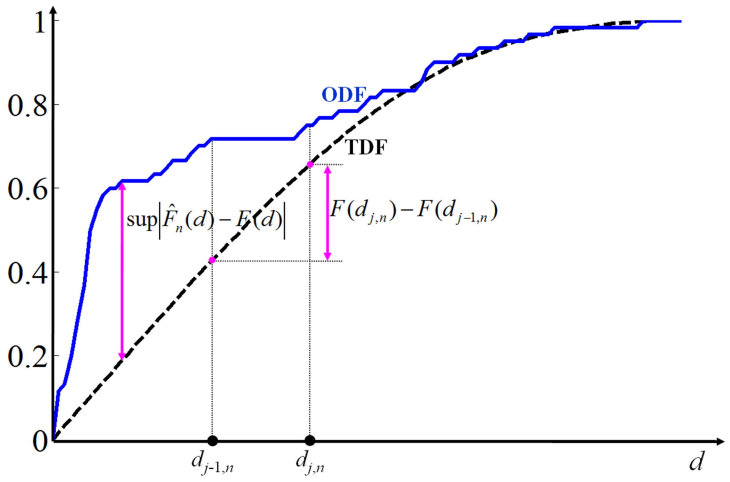


Figure 3.11: Illustration of K-S test.

$\mathbf{x}^1 = \{x_1^1, \cdots, x_n^2\}$ and $\mathbf{x}^2 = \{x_1^2, \cdots, x_m^2\}$:

$$\begin{aligned} &H_0 : \mathbf{x}^1 \text{ and } \mathbf{x}^2 \text{ are from the same distribution.} \\ &H_A : \mathbf{x}^1 \text{ and } \mathbf{x}^2 \text{ are not from the same distribution.} \\ &D_2 = \sup_d |\hat{F}_n(d) - \hat{F}_m(d)| \\ &\text{Test statistic} : S_2 = \sqrt{\frac{nm}{n+m}} \cdot D_2, \end{aligned} \tag{3.35}$$

where $\hat{F}_n(d)$ and $\hat{F}_m(d)$ are the ODFs from $\mathbf{x}^1$ and $\mathbf{x}^2$, respectively. The critical values for the K-S test are well studied and can be explicitly obtained [72, 73]. According to these studies, the critical values for the test statistics under $90\%, 95\%$ and 99% confidence level are 1.22, 1.36 and 1.63, respectively.

4 CONTAMINATION SOURCE IDENTIFICATION BASED ON SEQUENTIAL BAYESIAN APPROACH FOR WATER DISTRIBUTION NETWORK WITH STOCHASTIC DEMANDS*

Efficient anomaly detection is crucial to the safe operation of smart and connected systems. In this chapter, we choose the contamination source identification problem in water distribution network to demonstrate the efficient anomaly detection technique in network based systems. In the water distribution network, sensor alarms are recorded in multiple simulations to establish the observation probability distribution function. Then the observation probability distribution information is used to compute the posterior probability of each possible source for the observed alarm pattern in real time. Finally the contamination source is identified based on the ranking of the posterior probability. The critical contribution of this work is that the probability distributions for all possible observations are organized into a concise hierarchical tree structure and the challenge of combinatorial explosion is avoided. Furthermore, the variation analysis towards the posterior probability is conducted to give significance probability to the identification result we obtained. The effectiveness of this method is verified by a case study with a realistic water distribution network.

4.1 Introduction

Urban water distribution network is an important part of civil infrastructure that provides clean water and ensures quality of life. A water distribution network usually consists

*This chaper is based on the paper: **Wang C.** and Zhou S. (2017). "Contamination Source Identification Based on Sequential Bayesian Approach for Water Distribution Network with Stochastic Demands". *IISE Transactions*, 49 (9): 899-910

of water consumption locations, such as water taps at home (referred as nodes), pumps, pipes and storage tanks [80]. The water distribution network is vulnerable to a range of threats including direct physical attacks on the water supply infrastructure, sabotage of supervisory control and/or data acquisition instruments, and chemical or biological contamination [81]. Thus, a warning system is important to protect water distribution networks from possible pipe breaks and accidental or intentional contaminant events [82]. Figure 4.1 is an illustration of a simple water distribution network with two detection sensors. The network includes a reservoir (e.g. a treatment plant), six consuming nodes, a storage tank and several pipes connecting the nodes. The water is pumped from the reservoir into the network for the consumption by each node. The water flow rate and direction within pipes are determined by factors such as the pump power, water demands at nodes, and the tank volume. The flow rate and direction along time are called hydraulic characteristics of a water distribution network, which can be calculated by iteratively balancing the hydraulic energy getting in and out of the node [83].

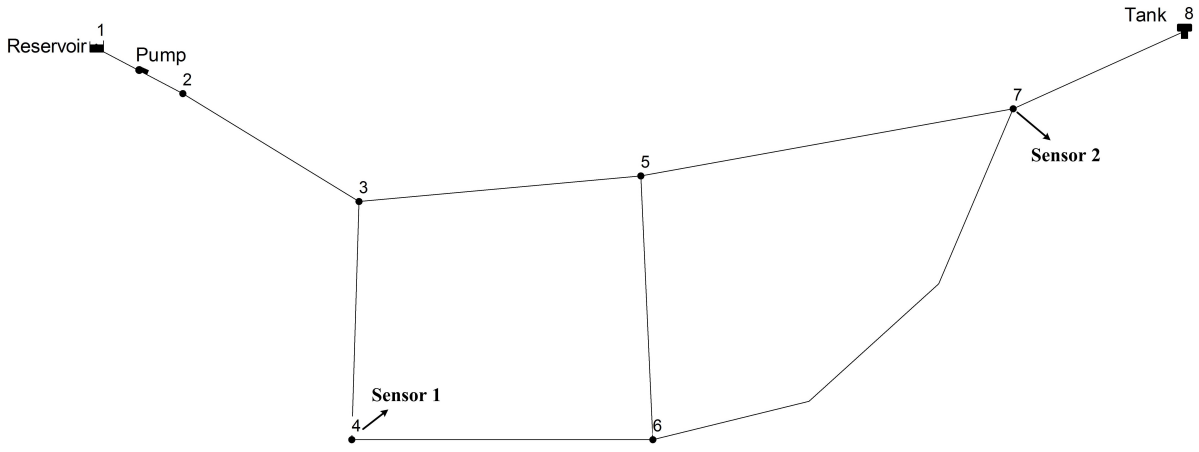


Figure 4.1: Diagram of water distribution network with sensors.

The sensors allocated in node 4 and node 7 in Fig. 4.1 serve to detect the concentration and/or existence of contamination materials, such as chlorine. The contamination is

identified by four parameters: location (e.g. the node of contamination), time (starting time instant and duration), intensity, and material. Once the contamination happens, the contamination material released from the source node will flow with water and reach the accessible nodes as time goes by. In a warning system, the sensor readings are periodically collected. If the warning system detects contamination at one or more sensor locations, algorithms should be carried out to determine the four parameters for contamination. Particularly, the contamination location should be identified first to isolate the source hydraulically from the large network, so that it is possible to limit the total contamination mass added [84]. In this article, we will focus on the problem of identifying the location (i.e., node) of the contamination source.

Many methods have been proposed to identify the contamination source in the water distribution network. These methods can be roughly put into two categories: deterministic methods and stochastic methods. The deterministic methods assume the water demands at each node are deterministic and all the hydraulic characteristics in the water distribution network are known at each time instant, which means the water flows and directions in pipes have a fixed pattern that can be predicted. The stochastic methods treat all the water demands at each node as random variables so that the water flows and directions in pipes are stochastic and cannot be exactly predicted.

Among the deterministic approaches, Laird et al. framed the contamination source identification problem into a least-square formulation, where the contamination source is identified by minimizing the sum of squares of the difference between the simulated and measured contamination concentration detected by sensors [85, 86]. They constructed the objective function by comparing the hydraulic characteristics with and without contamination injection terms (i.e., the location and injection time of the contamination source). This method depends on accurate simulation of the hydraulic characteristics of the network.

However, in practice, the simulation is very time consuming and thus the optimization step often needs a large amount of time to get a sufficiently accurate result, which is impractical for the urgent remedy for water distribution networks [87]. Further, this method needs relatively accurate contamination concentration readings [88]. To tackle this problem, De Sanctis et al. [84] employed binary sensor data (i.e., reading of normal vs abnormal) and used Particle Backtracking Algorithm (PBA) [89] to compute the contamination source weight at each node and then determine the candidate contamination sources in real time based on the weights. PBA computes the inverse propagation of contamination based on the hydraulic characteristics, which works from the alarmed sensors back to upstream nodes. PBA can give the contamination weights contributed from every upstream node to specified downstream nodes. Once the weight goes beyond a certain threshold, the node status will be updated. The number of sample period and the corresponding node status at that time are then used for the contamination source identification. This work was further extended with Bayesian probabilistic approach to consider false positives and false negatives in sensor readings [90]. Due to the efficient PBA, the method in [84] can quickly get the candidate contamination locations and injection time. Similar works for considering the measurement noise can be found in [91, 92, 93] with Bayesian filter. The PBA based method proposed in [84] and the optimization oriented method introduced in [85, 86] were extended in a few later works [94, 95, 96, 97, 98]. However, the deterministic approaches assume the hydraulic characteristics to be deterministic and precisely known, which is restrictive in practice.

The stochastic methods relax the restrictive assumptions of deterministic parameters in the water distribution network and treat the water demands, the water flow rates and directions in the pipes as random variables. The stochastic methods are mainly data-driven, which focus on the modeling of the sensor alarm information rather than

the hydraulic process of water flows. Bayesian approach has been a popular approach for monitoring the stochastic data given its capacity for providing probabilistic inference and incorporating prior knowledge [99, 100], thus draws attention to the application in the water distribution network with stochastic demands. Dawsey et al. first applied Bayesian belief networks (BBN) to the monitoring and contamination source identification of water distribution networks [101]. They classified nodes into several clusters so that water flows among clusters can construct a directed graph. Then, the BBN analysis is performed with prior information about the water distribution network to identify the possible clusters that contain contamination sources. This BBN method can deal with the stochastic water demand to some extent. However, when the cluster size becomes too big due to large demand variances, the identification result will not be accurate enough to locate the contamination node. Wang and Harrison utilized Bayesian analysis and data mining techniques to estimate the contamination source information given the sensor alarm observations [102]. The randomness of water demands is transferred to the sensor alarm observations, then the Bayesian analysis is employed to deal with the observations. However, in their model, only one sensor in the water distribution network is allowed, which leads to identifiability issues. Furthermore, the likelihood function in the Bayesian analysis can only be obtained by running a large number of simulations. Wang and Harrison further improved their method by using support vector regression to reduce the computation load [103]. Nonetheless, the improved method still suffers from the low identifiability and accuracy problems, which is caused by the assumption that only one sensor is in the water distribution network.

Considering the limitations of the existing methods, we propose a sequential Bayesian method for contamination source identification in water distribution network with stochastic demands. The basic idea of the work is relatively simple. In offline mode, we first obtain the

probability distributions of the sensor observations given a node being the contamination node through simulations. Then in real time, we compute the probability (also called posterior probability) of the node being the contamination node given the current sensor observations. If we go through this process for each node in the network, we can compare the posterior probability of each node and select the node with the highest posterior probability as the contamination node in real time. The idea is simple but one critical challenge is how to obtain an efficient way to represent the probability distribution of a large number of types of the observations due to the combinatorial explosion and compute the posterior probability. The main contribution of this work is that through the factorization of the posterior probability, we establish a concise hierarchical "tree" type of structure to represent the observation probability distribution. The hierarchical tree is a commonly used method in modeling the hierarchical nature of a structure in a graphical form. It was widely used in decision making [104], image processing [105] and classification [106].We propose to employ the hierarchical tree structure to model the observation probability distribution so that the posterior probability of the node being contamination node can be easily computed. Another highlight of our method is that the posterior probability calculation is conducted in a sequential way along with the arrival of the observations. In other words, as soon as we receive the first sensor alarm, we can start to update the posterior probability for each node using the Bayesian approach. Then each time we get a new observation (i.e., either a new alarm at another sensor or no alarm at all), the posterior probability is updated. Further, not only the information of which sensors alarm but also the time interval between sensors are taken into consideration. The effectiveness of proposed approach is illustrated with a realistic municipal water distribution network.

The rest of this chapter is organized as follows. Section 4.2 formulates the problem with some assumptions and notation definitions. The proposed sequential Bayesian update

approach is introduced in Section 4.3, where the procedures to obtain the observation probability distribution function and the variation analysis of the result are also presented. Section 4.4 presents a case study based on EPANET simulation to show the effectiveness of our method. Section 4.5 draws conclusion remarks.

4.2 Problem formulation

In a water distribution network, the contamination may happen at any node and at any time. We want to identify the location of the contamination source after at least one sensor alarms. In our proposed method, we have the following assumptions:

A6 There is only one contamination node during the identification time period. This assumption will significantly reduce the computational load.

A7 The contamination happens as a step input, which means once the contamination injected into one of the nodes, it will keep releasing contamination at that node. This assumption is consistent with the fact that the typical water contamination resulted from chemical leakage or biological pollution will continue or even become worse until remedial actions are implemented [107].

A8 Water demands at each node have truncated Gaussian distribution with specified mean and variance. The demands below zero are truncated from the Gaussian distribution [94, 108].

A9 All sensors are binary type with readings of normal/abnormal. The noise in the senor readings are ignored. Further, all the sensor readings are sampled at the same time instant and with the same sample interval. Once the sensor alarms, it keeps alarming until intervention is conducted.

We adopt these assumptions mainly due to the need of the consistency with system physics, computational load reduction, and system identifiability considerations. In fact, these assumptions are widely used in existing literatures and realistic in real water distribution network [102, 109].

We define the following notations to mathematically formulate the contamination source identification problem.

Notations:

-$C(k)$: Until the kth observation, the number of sensors that have been triggered.

-K_i: Observation index that represents the first time i sensors have alarmed.

$$K_i = \inf(k \geq 0 : C(k) = i) \tag{4.1}$$

Then, the alarming time instant for n sensors is represented by $K(n)$:

$$\mathbf{K}(n) = \{K_1, K_2, \cdots, K_n\}, 1 \leq n \leq I \tag{4.2}$$

where $I = \max(C(k))$ represents the total number of alarmed sensors until the kth observation.

- $S(k)$: At the kth observation, the most recently alarmed sensor.

Then, the alarmed sensor sequence until the nth sensor alarms can be given by $\mathbf{S}(n)$:

$$\mathbf{S}(n) = \{S(K_1), S(K_2), \cdots, S(K_n)\}, 1 \leq n \leq I \tag{4.3}$$

With the above definitions, the information about sensor alarm until the time instant

K_i (the time instant when the ith sensor alarm) can be represented by $\mathbf{Y}(i)$:

$$\mathbf{Y}(i) = \{\mathbf{K}(i), \mathbf{S}(i)\}, 1 \leq i \leq I \tag{4.4}$$

We further define the random variable X as the number of observations between the current observation instant and the most recent alarm time,

$$X = k - K_I \tag{4.5}$$

$\mathbf{Y}(I)$ and X fully represent the observation sequence until the kth observation: the observation has sensor alarm sequence $\mathbf{S}(I)$ with corresponding alarm time $\mathbf{K}(I)$ followed by X non-alarm observations.

To illustrate how these notations work, we give a simple example in Table 4.1, where there are 3 sensors in the water distribution network. To indicate the sensor alarm status, we use '*1*' to indicate abnormal reading, whereas '*0*' as normal reading.

In Table 4.1, the sensor 1, 3, 2 alarms in sequence at the observation instant 1,k and n, which results in $I = 1, I = 2$ and $I = 3$ at corresponding alarm instants. The alarm information $\mathbf{Y}(i) = \{\mathbf{K}(i), \mathbf{S}(i)\}$ can be obtained by making (K_i, i) and $(S(k), i)$ columns into vectors through Equation 4.2 and 4.3.

If we let a random variable z represent the contamination source location and $P_0(z)$ as the prior probability mass function (pmf) of z, then the contamination source identification problem can be formulated as the problem of computing the posterior distribution of z given all the observed alarm information, $P(z|\mathbf{Y}(I), X)$. Those nodes with high posterior probabilities can be picked as potential contamination nodes.

Table 4.1: Illustration for notations.

Sensor index			Observation index k	$C(k)$	(K_i, i)	$(S(k), i)$	I	X
1	**2**	**3**						
1	*0*	*0*	1	1	$(1,1)$	$(1,1)$	1	0
1	*0*	*0*	2	1	$(1,1)$	$(1,1)$	1	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	*0*	*0*	$k-1$	1	$(1,1)$	$(1,1)$	1	$k-1$
1	*0*	*1*	k	2	$(k,2)$	$(3,2)$	2	0
1	*0*	*1*	$k+1$	2	$(k,2)$	$(3,2)$	2	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	*1*	*1*	n	3	$(n,3)$	$(2,3)$	3	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

4.3 Sequential update for $P(z|\mathbf{Y}(I), X)$ using Bayesian approach

With notations defined in Section 4.2, we can utilize Bayesian formula to compute $P(z|\mathbf{Y}(I), X)$ as follows:

$$P(z|\mathbf{Y}(I), X) = \frac{P(\mathbf{Y}, X|z) \cdot P_0(z)}{\sum_z P(\mathbf{Y}, X|z) \cdot P_0(z)} \tag{4.6}$$

The numerator of Equation 4.6 consists of the likelihood term $P(\mathbf{Y}(I), X|z)$ and the prior distribution $P_0(z)$. Without the loss of generality, we can set the prior $P_0(z)$ as a noninformative uniform multinomial distribution. The denominator is a normalization constant to ensure the posterior distribution sums to unity. Since we mainly concern about

which node is the most probable contamination source rather than the exact value of the posterior probability, we only need to consider the numerator term. Thus, we can have the problem simplified as follows:

$$P(z|\mathbf{Y}(I), X) \propto P(\mathbf{Y}(I), X|z) \cdot P_0(z) \tag{4.7}$$

Since the $P_0(z)$ is set as uniform (i.e., each node has the same prior probability), we can only consider $P(\mathbf{Y}(I), X|z)$:

$$P(\mathbf{Y}(I), X|z) = P(X|\mathbf{Y}(I), z) \cdot P(\mathbf{Y}(I)|z) \tag{4.8}$$

Equation 4.8 has two terms. For $P(X|\mathbf{Y}(I), z)$, with the law of total probability, we have:

$$P(X|\mathbf{Y}(I), z) = \sum_{S(K_{I+1})=0}^{m} P\Big(X|z, \mathbf{Y}(I), S(K_{I+1})\Big) \cdot P\Big(S(K_{I+1})|z, \mathbf{Y}(I)\Big) \tag{4.9}$$

where m is the total sensor number, and $S(K_{I+1}) = 0$ represents the situation of no further alarm since I alarms happen.

If we define $f_{T|\mathbf{Y}(p),z}(T)$ be the pmf of time T between two consecutive sensor alarms (i.e., the $(p-1)$th and the pth alarms) given $\mathbf{Y}(p)$ and z, and noting that $P\Big(X|z, \mathbf{Y}(I), S(K_{I+1})\Big)$=1$-$ $\sum_{T=1}^{X} f_{T|\mathbf{Y}(I+1),z}(T)$, then Equation 4.9 can be further expanded as:

$$P(X|\mathbf{Y}(I), z) = \sum_{S(K_{I+1})=0}^{m} \Big(1 - \sum_{T=1}^{X} f_{T|\mathbf{Y}(I+1),z}(T)\Big) \cdot P\Big(S(K_{I+1})|z, \mathbf{Y}(I)\Big) \tag{4.10}$$

For the second term $P(\mathbf{Y}(I)|z)$ in Equation 4.8, we have

$$P(X|\mathbf{Y}(I),z)=\sum_{S(K_{I+1})=0}^{m}\Big(1-\sum_{T=1}^{X}f_{T|\mathbf{Y}(I+1),z}(T)\Big)\cdot P\Big(S(K_{I+1})|z,\mathbf{Y}(I)\Big) \tag{4.11}$$

Equation 4.11 can be derived as follows:

$$\begin{aligned}P(\mathbf{Y}(I)|z)&=P\Big(\mathbf{Y}(I-1),K_I,S(K_I)|z\Big)\\&=P\Big(K_I|z,\mathbf{Y}(I-1),S(K_I)\Big)\cdot P\Big(S(K_I)|z,\mathbf{Y}(I-1)\Big)\cdot P\Big(\mathbf{Y}(I-1)|z\Big)\\&=f_{T|\mathbf{Y}(K_I),z}(K_I-K_{I-1})\cdot P\Big(S(K_I)|z,\mathbf{Y}(I-1)\Big)\cdot P\Big(\mathbf{Y}(I-1)|z\Big)\end{aligned} \tag{4.12}$$

where the term $P(Y(I-1)|z)$ can be iteratively calculated through the process from Equation 4.12. Then we can get the result in Equation 4.11. Combining Equation 4.10 and 4.11, we can have the final posterior probability expression as:

$$\begin{aligned}&P(\mathbf{Y}(I),X|z)\\&=\Big(\sum_{S(K_(I+1))=0}^{m}\Big(1-\sum_{T=1}^{X}f_{T|Y(I+1),z}(T)\Big)\cdot P\Big(S(K_{I+1})|z,\mathbf{Y}(I)\Big)\Big)\cdot\\&\prod_{q=2}^{I}\Big(f_{T|\mathbf{Y}(q),z}(K_q-K_{q-1})\cdot P\Big(S(K_q)|z,\mathbf{Y}(q-1)\Big)\Big)\cdot P\Big(S(K_1)|z\Big)\end{aligned} \tag{4.13}$$

Equation 4.13 shows the computing procedure for $P(\mathbf{Y}(I),X|z)$ and the information we will need in the calculation. To give an intuitive understanding of the procedure, we provide an illustrative example. Assume we have a water distribution network with 3 sensors and we want to check if node 7 is the contamination node. The terms we will need to evaluate Equation 4.13 is shown in Fig. 4.2. The first row in Fig. 4.2 shows the distribution of the first alarmed sensor given $z=7$, i.e.,$P\Big(S(K_1)|z=7\Big)$, where horizontal axis represents the sensor index and the number "4" means no sensor alarms (please note that it is possible that no sensor is alarmed even if node 7 is contaminated). The

probability of the next alarming sensor $P\Big(S(K_2)|z=7,\mathbf{Y}(1)\Big)$ is marked on the arrow line. The corresponding distribution of the time from the first alarm $\Big(f_{T|\mathbf{Y}(2),z=7}(T)\Big)$ is shown in the second row of Fig. 4.2. Similarly, the probabilities and distributions corresponding to the third alarm are shown in the third row. With this information, the posterior probability of $P(z=7)$ given the observed alarm sequence can be easily computed. For example, if the observed sensor alarm sequence is $1 \to 3 \to 2$ with 2 time units between 1 and 3 and 3 time units between 3 and 2, then $P\Big(z=7|\mathbf{Y}(3)\Big)$ can be computed by multiplying the probabilities circled in Fig. 4.2 according to Equation 4.13. For an intermediate observation, the posterior probability of $P(z=7)$ can also be computed. For example, if we have observed the alarms from sensor 1 and 3 and one time unit has passed after the alarm at sensor 3, i.e., $\mathbf{Y}(2)=\{(0,2),(1,3)\}$ and $X=1$, then $P\Big(z=7|\mathbf{Y}(2),X\Big)$ is to multiply the circled probabilities in the first two rows and the term $\Big(1-f_{T|\mathbf{Y}(3),z=7}(T=1)\Big)\cdot P\Big(S(K_3=2)|z=7,\mathbf{Y}(2)\Big)+\Big(1-P\Big(S(K_3=2)|z=7,\mathbf{Y}(2)\Big)\Big)$ according to Equation 4.13, where $\mathbf{Y}(3)=\{(0,2,1),(1,3,2)\}$.

Figure 4.2 illustrates how we obtain the posterior probability for node 7 being the contamination node given the alarm observations. Intuitively, if we have the required probability distribution information as shown in Fig. 4.2 for other nodes (we will discuss how we can obtain such information through simulation in the following section), we can compute the posterior probability for other nodes being the contamination node. By comparing the posterior probabilities, we can determine the possible contamination node.

Specifically, we define

$$\begin{aligned} z^1 &= \arg\max_z \Big(P(z|\mathbf{Y}(I),X)\Big) \\ z^i &= \arg\max_{z\neq z^1,z^2,\cdots,z^{i-1}} \Big(P(z|\mathbf{Y}(I),X)\Big) \end{aligned} \tag{4.14}$$

where z^1 represents the node with the highest posterior probability and z^i is the node

with the ith highest posterior distribution. Then we can use the rule

$$z \in \{z^1, z^2, \cdots, z^G\} \tag{4.15}$$

to estimate the contamination node, where G is a selected small positive integer.

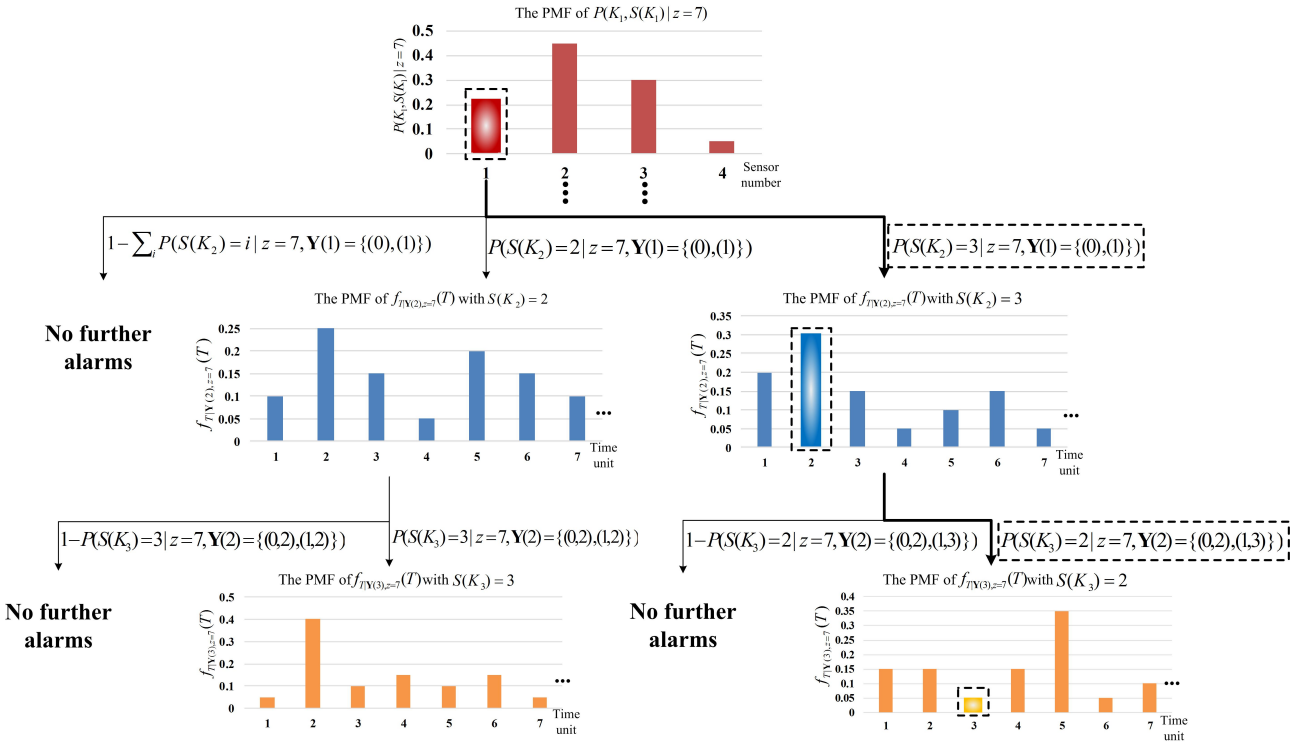


Figure 4.2: Illustration of computing posterior of $z = 7$.

4.3.1 Monte Carlo simulation procedure for estimating observation probability distribution

In the posterior probability updating procedure, we need the pmf term $f_{T|\mathbf{Y}(p),z}(T)$ and weights term $P\Big(S(K_{I+1})|z, \mathbf{Y}(I)\Big)$ in Equation 4.13. The values of these two terms can be

obtained through a Monte Carlo simulation procedure. The basic idea of Monte Carlo simulation is to conduct a large number simulation runs of the hydraulic characteristics of the water distribution network for a given contamination node. For each run, we will obtain simulated sensor alarm observations. Then we can count the frequencies of the alarm observations and from which, we can obtain the required observation probability information. We first define two counting variables to help record different alarming situations.

(1) $D_u, u = 1, 2, \cdots, m+1$ is defined as the number of simulation runs that the first alarm is by sensor u, where m is the total sensor number and $m+1$ represents the situation that no alarm occurs.

(2) $F_(Y(v), K_v - K_(v-1)), v = 2, 3, \cdots, d$ is defined as the number of simulation runs that the alarm $\mathbf{Y}(v)$ is observed with the time between two most recent alarms being $K_v - K_{v-1}$, where d is the total number of alarmed sensors in the simulation run. Please note that $\mathbf{Y}(v)$ and $K_v - K_{v-1}$ should exhaust all the possible alarm patterns and thus we have a large number of $F_{\mathbf{Y}(v),K_v-K_{v-1}}$s.

With the defined counting variables, the detailed simulation steps are as follows:

1) Initiating all the counting variables to 0 and set the contamination node $z = 1$.

2) Randomly select a contamination time and inject the contamination and generate the water demand in each node using the truncated Gaussian distribution.

3) Run EPANET to calculate the hydraulic characteristics and the contamination propagation in the network.

4) Record the sensor alarm sequence and time and update the counting variables accordingly.

5) Repeat step 2) to 4) N times to get the final value of the all the counting variables.

6) Setting the contamination node to a different node and repeat Step 1) to 5). This process should be conducted for each node in the network.

With the counting variable values for each contamination node, we can estimate the terms in Equation 4.13 as follows:

$$\hat{P}\Big(S(K_1)|z\Big) = \frac{D_{S(K_1)}}{\sum_{u=1}^{m+1} D_u}, S(K_1) = 1, 2, \cdots, m+1 \tag{4.16}$$

$$\hat{f}_{T|\mathbf{Y},z}(T) = \frac{F_{\mathbf{Y}(v),T}}{\sum_r F_{\mathbf{Y}(v),r}}, T = 0, 1, 2, \cdots \tag{4.17}$$

To compute Equation 4.13, we also need $P\Big(S(K_{v+1}|z, \mathbf{Y}(v)\Big)$, which is the probability for observing alarm from sensor $S(K_{v+1})$ after we already observed alarm information $\mathbf{Y}(v)$,

$$\hat{P}\Big(S(K_{v+1})|z, \mathbf{Y}(v)\Big) = \frac{\sum_r F_{\mathbf{Y}(v+1),r}}{F_{\mathbf{Y}(v),r}} \tag{4.18}$$

In one simulation for the specific contamination node, the no alarm count in Equation 4.16 is determined by observing no sensor alarm at the end of the simulation, which is affected by the simulation duration and the contamination injection time. The simulation repetition number N is related with the network size, demands variance, sensor location and accuracy of the result. In practice, people can use trial and error method to determine the parameter N to get desired result accuracy for specific network and warning system. Another point we would like to mention is that although the Monte Carlo simulations for obtaining the observation probability distribution function consumes time, it is an

offline process and can be carried out concurrently on many computers. Once this process finishes, the contamination source identification process based on the observation probability distribution function is fast.

4.3.2 Variation analysis of posterior probability

From Section 4.3.1, we can see that the observation probability distribution function is estimated based on a finite simulation runs of the underlying multinomial distributions. Thus, observation probability distribution function contains variations, which in turn leads to the variations in the posterior probability for a given node and a specific observation. The decision making rule (Equation 4.15) is determined by the ranking of the posterior probabilities. In this section, we would like to investigate uncertainty in the obtained order of the posterior probability. Specifically, we would like to obtain the probability

$$P\Big(\min(\hat{p}^1, \cdots, \hat{p}^G) > \max(\hat{p}^i, i = G+1, \cdots, l)\Big) \tag{4.19}$$

where $\hat{p}^i = \hat{P}(z = i|\mathbf{Y}(I), X)$ and i is the ranking of the posterior probability, l is the total number of node in the network, the symbol $\hat{}$ indicates that the value is an estimated value. The probability in Equation 4.19 indicates how confident we are regarding the identified group of nodes. Considering the independence among $\hat{p}^i$s, Equation 4.19 can be transformed to:

$$\prod_{H=1}^{G} \prod_{s=G+1}^{l} P(\hat{p}^H > \hat{p}^s) \tag{4.20}$$

To compute the probability, we will need the probability distribution of estimated posterior probability $p^i, i = 1, \cdots, l$. However, the analytical form the distribution is not tractable. Here we propose an approximation approach.

(1) In this first step, we calculate the mean and variance of the estimated

posterior probability of each node. Fortunately, through a tedious analysis of Equation 4.8, the variance of the estimated posterior probability can be obtained analytically as a function of number of simulation runs. Due to the long expression, the result is presented in the appendix. Readers interested in the derivation procedure can contact authors.

(2) In the second step, we will approximate the distribution of the estimated posterior probability of each node by fitting a Beta distribution with the mean and variance. Beta distribution is very flexible and is widely used to represent the distribution of a probability [110]. A Beta probability density function is determined by two parameters α and β. If the mean (μ) and variance (σ^2) of the Beta random variable is known, then the distribution parameters can be obtained as

$$\alpha = \frac{\mu^2}{\sigma^2} - \frac{\mu^3}{\sigma^2} - \mu \tag{4.21}$$

$$\beta = \frac{\alpha}{\mu} - \alpha \tag{4.22}$$

(2) With the fitted Beta distribution function, we can easily obtain the terms in Equation 4.20. We define this probability as significance probability.

We compared this proposed procedure with a computational intensive bootstrapping procedure in the following case study section.

4.4 Case study

In the case study, we adopt the popular simulation software EPANET, which can simulate extended period of hydraulic and water quality behavior within pressurized pipe networks

[111]. EPANET can track the flow of water in each pipe, the pressure at each node, the height of water in each tank, and the concentration of a chemical species throughout the network during a simulation period comprised of multiple time steps. The water distribution network in the case study is from [112], which includes 126 nodes, 168 pipes, one constant head source, two tanks, two pumps, and eight valves. The whole network is shown in Fig. 4.3. All the hydraulic parameters and water demand patterns for each node are left unchanged. The detailed parameter values can be found in the study of [113].

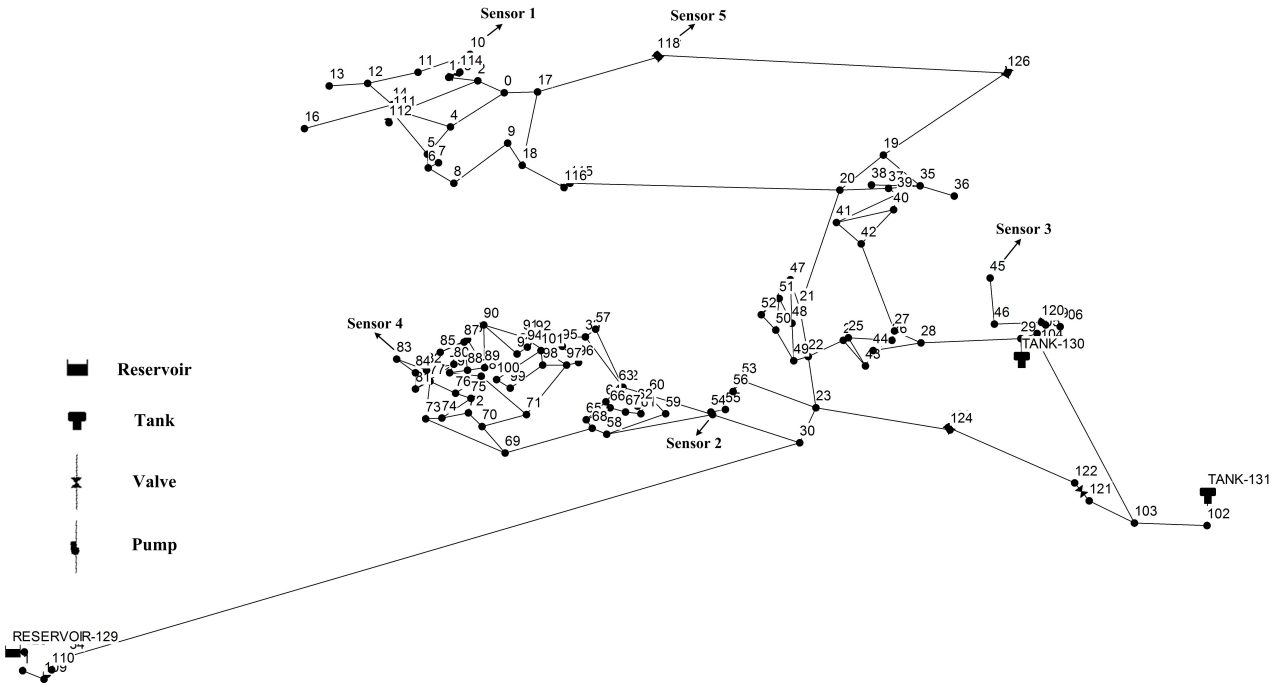


Figure 4.3: Layout of the case study water distribution network.

In this case study, the simulated duration is 48 h with hydraulic time step 0.5 h and water quality time step 10 min (which means we sample the sensor readings every 10 min). The contamination injection is set as 1.0 kg/min at a random hydraulic period between 0 and 24 h. The contamination can reach any accessible node within the 48 h simulation

period. Only one injection node is considered in each simulation run. The water demand at each node within each hydraulic period is set as the truncated normal with mean equal to the demand data in the study of [113]. A 40% coefficient of variation is applied at each of the truncated normal distribution to model the variance. This stochastic water demands setting has also been applied elsewhere in [102]. Total 5 sensors are placed in the system and the locations are shown in Fig. 4.3. For the offline information collection, at each node, we conduct $N = 3000$ times of repeated simulation runs to obtain the needed observation probability distribution function. Then, a node is randomly selected as the contamination node and the alarm observations are simulated. Based on the alarm observations and the observation probability distribution function, we can use the proposed Bayesian approach to identify the contamination node and check if it is correct.

Specifically, we illustrate an example of how the contamination source identification is conducted. In this example, we inject the contamination at node 20 at a randomly chosen injection time. We first observe the alarm at sensor 5, then sensor 1 alarms after two time units, and then sensor 3 alarms after seventeen time units. No further alarms are observed. Following the posterior probability calculation procedure, we can compute the posterior probability for each node being the contamination node at each time unit. The posterior probabilities of top three nodes are shown in Fig. 4.4(a). The true contamination node, represented by the line with stars in Fig. 4.4(a), ranks the highest at most of time during the identification process and finally stands out, which shows the effectiveness of our proposed method. We can also observe the downward trend of the posterior probability of the true contamination node and the upward trends of that of two different nodes (node 19 and node 21 in Fig. 4.3) between the second and the third sensor alarms. This is because node 19 and node 21 are close to node 20, thus their observation probability distribution functions are similar in some parts. To address this problem, we can use

the variation analysis technique introduced in Section 4.3.2 to obtain the significance probability of the ranking of the posterior probabilities. In Fig. 4.4(b), we show the probability $P\Big(\min(p^1) > \max(p^i, i = 2, \cdots, l)\Big)$, the probability that the node with the highest posterior probability is indeed the node with the highest posterior probability, in Fig. 4.4(b) by the line with solid circles. We can see that the probability drops dramatically between the second and the third alarms and keeps under 50% before the third alarm happens. This means that we cannot make confident decisions about which node is with the highest posterior probability. Under this situation, we should include more nodes as the candidate contamination nodes or wait for more observations to make decision.

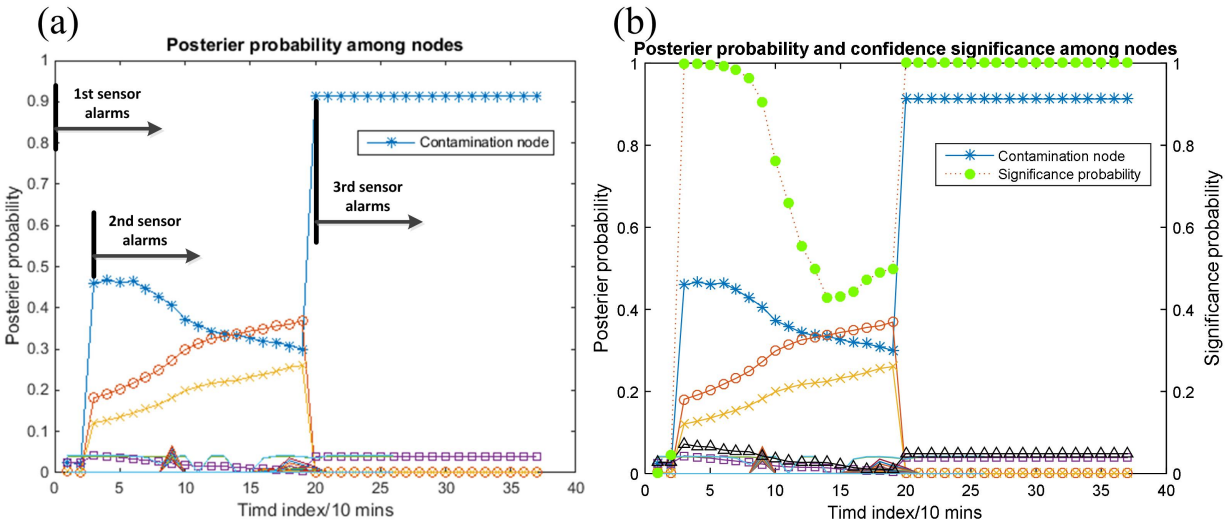


Figure 4.4: Posterior probability and significance probability

We further evaluated the effectiveness of the proposed method of computing the significance probability based on Beta distribution fitting through a comparison with a computational intensive bootstrapping method. Bootstrapping is a commonly used resampling technique to obtain uncertainty measures on estimated statistical values [114]. In Fig. 4.5, we plotted the significance probability obtained from bootstrapping and

the proposed method based on Beta distribution fitting. There is little difference in the comparison results. Please note that the proposed method can be computed through a closed form expression thus requires much less computational load comparing with bootstrapping method. Thus, the proposed Beta distribution fitting method is a fairly effective and efficient method for significance probability evaluation.

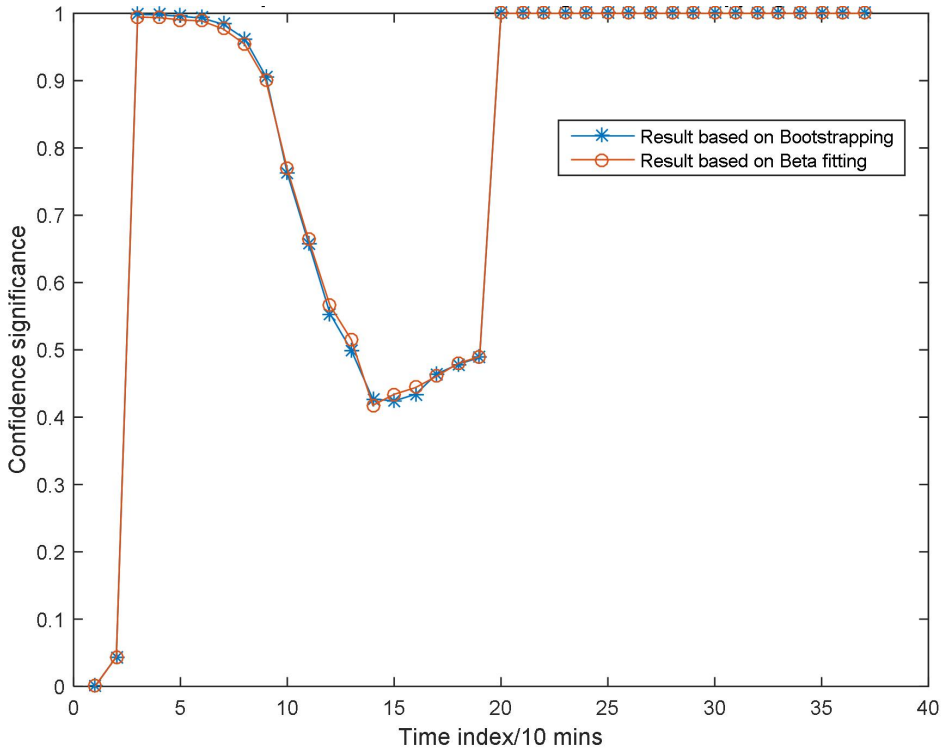


Figure 4.5: Comparison between Beta fitting method and Bootstrapping method

Above discussion illustrates the sequential contamination node identification procedure along the arrival of observations. To evaluate how well the procedure can identify the true contamination node, we conduct 500 times of cases of simulation for each node being the contamination node and randomly choosing the contamination time. In 48.1% of these 63000 (126×500) verification cases, the true contamination node can be correctly identified if we use $G = 1$ in the decision rule, i.e., the node with the highest posterior probability is

the contamination node. If we use $G = 3$ in the decision rule, i.e., the contamination node is among the top three nodes with the highest posterior probabilities, then in 70.2% of all the verification cases, the true contamination node can be correctly identified. In Fig. 4.6, we provide a visualization of how well each node can be identified. In Fig. 4.6, the larger the point indicates that the higher possibility that the node can be identified with the decision rule of $G = 3$. We can see that many nodes can be (almost) always identified, but some nodes can never be identified. Intuitively, the distribution of the nodes that can be easily identified and the nodes that cannot be identified will be influenced by sensor number and sensor location. The reason that some nodes can never be identified is that the contamination from those nodes may not be able to reach the sensor locations.

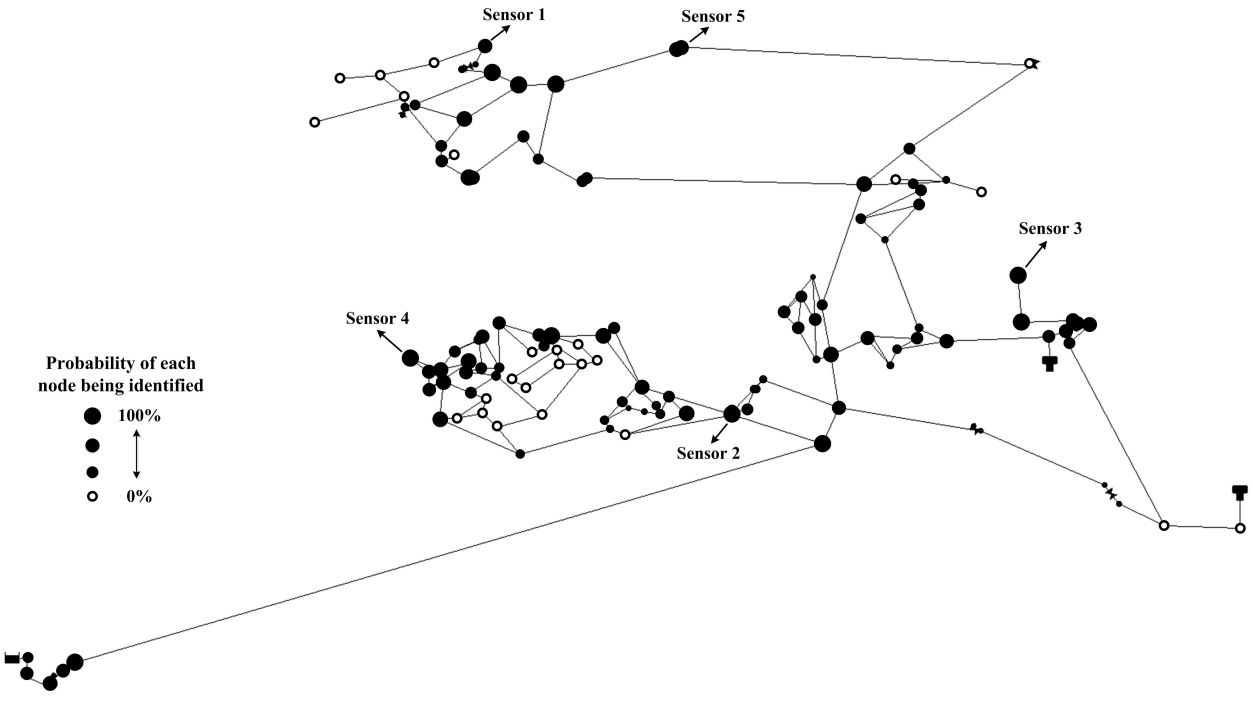


Figure 4.6: Probability of each node being correctly identified

We also investigate the relationship between sensor alarms number and the probability

that we can correctly identify the true contamination node. The result is shown in Table 4.2, where we give the percentage among all the verification cases that we correctly identify the true contamination node under different number of sensor alarms. Not surprisingly, we can see that with only one sensor alarm, it is difficult to identify the contamination node, while with more sensor alarms, we get higher percentages of correct identification.

Table 4.2: The identification ability evolves with sensor alarms.

Alarm number(s)	1	2	3	4	5
Identifiability	2.7%	17.6%	37.9%	52.5%	70.2%

The probability of correctly identifying the true contamination node is a very important measure of the effectiveness of an identification method. A correct identification rate of 48.1% for $G = 1$ and 70.2% for $G = 3$ seem not high. However, if we consider there are 126 nodes in the system, the water demand is stochastic, and the sensor readings are simple binary readings, the result is quite impressive. Indeed, a noninformative identification procedure (i.e., randomly pick G nodes as the contamination node) can only reach a correct identification probability of 0.8% for G=1 and 2.4% for G=3. The improvement over the noninformative method is quite significant. We also compared our performance under G=1 with that of several existing works, which is shown in Table 4.3. The benchmark methods are from [109] and [84]. In [109], the sensor readings at each time are regressed through logistic regression to indicate the probability of sensor alarming at each time for a specific contamination location. In [84], De Sanctis et al. utilized Particle Backtracking Algorithm (PBA) to compute the contamination source weight at each node and then determine the candidate contamination sources in real time based on the weights. We apply our method into the water distribution network in [109] and [84], respectively. We can see that our method outperforms in the probability of correct identification. This shows the effectiveness and accuracy of our method, which mainly because we use both

the sensor alarm sequence and time between successive alarms so that we can use more available information.

Table 4.3: Comparison of the correct identification probability among different methods.

Methods (sensors,nodes)	Liu et al. $(4, 97)$	De Sanctis et al. $(6, 97)$	De Sanctis et al. $(8, 97)$
Identifiability(Benchmark)	$< 27\%$	$< 20\%$	$< 21\%$
Identifiability(Proposed)	36.1%	39.2%	$< 40.2\%$

The computation load in our method includes the offline part for constructing the observation probability distribution function and the online part for updating posterior distribution and variation analysis. The offline part contains N times simulations for each node (126×N simulations in total in our case study), and each simulation consumes around 7 seconds. Although the offline computation load is large, it can be operated in parallel on many computers to reduce the time. The online part including the real time updating and the variation analysis takes only 50 seconds for each new observation, which means the decision of contamination node identification can be made within 1 min. This satisfies the real time warning system with 10 min or 5 min sensor reading period, and provides potential extension to larger water distribution networks. Our method operates with a PC 3.3 GHz, 32 GB RAM on Matlab 2014b.

4.5 Conclusion

In this chapter, we propose a sequential Bayesian scheme for contamination source identification in the water distribution network. This method treats the sensor alarm as available information and takes advantage of alarm sequence and time delay between alarms to do real time contamination source identifications. The posterior probabilities of each node being the contamination source is calculated and used for decision making. Moreover, the

variation analysis of the obtained posterior probability is conducted to give significance probability of the result we get. Our method considers the hydraulic information and stochastic water demands in the water distribution network, which is less restrictive in practice. The effectiveness of the method is verified by a case study in EPANET with a 126 nodes water distribution network.

The proposed method can be further explored in several directions. First, the computational load in our method will increase exponentially as the number of contamination parameters increases, which will limit the application of the proposed method in the multi-parameter identification case. Studies on reducing the computational load in order to extend the proposed method need to be done. Another interesting area is the influence of sensor layout on the result of contamination source identification. The sensor layout that includes the sensor number and sensor location needs to be optimized. We intend to work on these problems in the future.

4.6 Appendix

4.6.1 Variance of Equation 4.8

$$\hat{P}(\mathbf{Y}(I), X|z) = \Big(\sum_{S(K_{I+1})=0}^{m} \Big(1 - \sum_{T=1}^{X} \hat{f}_{T|\mathbf{Y}(I+1),z}(T)\Big) \cdot \hat{P}\Big(S(K_{I+1})|z, \mathbf{Y}(I)\Big)\Big) \cdot \Big(\prod_{q=2}^{I} \hat{f}_{T|\mathbf{Y}(q),z}(K_1 - K_{q-1}) \cdot \hat{P}\Big(S(K_q)|z, \mathbf{Y}(q-1)\Big) \cdot \hat{P}\Big(S(K_1)|z\Big)\Big)$$

Let

$$A = \sum_{S(K_{I+1})=0}^{m} \Big(1 - \sum_{T=1}^{X} \hat{f}_{T|\mathbf{Y}(I+1),z}(T)\Big) \cdot \hat{P}\Big(S(K_{I+1})|z, \mathbf{Y}(I)\Big)$$

$B = \prod_{q=2}^{I} \hat{f}_{T|\mathbf{Y}(q),z}(K_1 - K_{q-1}) \cdot \hat{P}\Big(S(K_q)|z, \mathbf{Y}(q-1)\Big) \cdot \hat{P}\Big(S(K_1)|z\Big)$ Then we have

$$\begin{aligned} \mathrm{Var}\Big(\hat{P}(\mathbf{Y}(I), X|z)\Big) &= \mathrm{Var}(A \cdot B) \\ &= \mathrm{E}^2(A) \cdot \mathrm{Var}(B) + \mathrm{E}^2(B) \cdot \mathrm{Var}(A) + \mathrm{Var}(A) \cdot \mathrm{Var}(B) \end{aligned} \tag{4.23}$$

where $\mathrm{Var}(A) = \mathrm{Var}(\sum_{i=0}^{m} A_i) = \sum_{i=0}^{m} \mathrm{Var}(A_i) + 2\Big(\sum_{i=0}^{m-1} \sum_{j=i+1}^{m} \mathrm{Cov}(A_i, A_j)\Big)$

$$A_i = \Big(1 - \sum_{T=1}^{X} \hat{f}_{T|\mathbf{Y}(I+1),z}(T)\Big) \cdot \hat{P}\Big(S(K_{I+1}) = i|z, \mathbf{Y}(I)\Big), 0 \le i \le m \tag{4.24}$$

Let

$\hat{P}_{\mathbf{Y}(i+1),j} = \hat{P}\Big(S(K_{i+1}) = j|z, \mathbf{Y}(i)\Big), 1 \le i \le I$

$\hat{p}_{\mathbf{Y}(i+1),j}(T) = \hat{f}_{T|\mathbf{Y}(i),S(K_{i+1})=j,z}(T), 1 \le i \le I$

Then, we have

$$\begin{aligned} &\mathrm{Var}(A_i) \\ &= \Big(1 - \sum_{T=1}^{X} \hat{f}_{T|\mathbf{Y}(I+1),z}(T)\Big)^2 \cdot \frac{\hat{P}_{\mathbf{Y}(I+1),i}(1 - \hat{P}_{\mathbf{Y}(I+1),i})}{\sum_r F_{\mathbf{Y}(I),r}} + \Big((\hat{P}_{\mathbf{Y}(I+1),i})^2 + \frac{\hat{P}_{\mathbf{Y}(I+1),i}(1 - \hat{P}_{\mathbf{Y}(I+1),i})}{\sum_r F_{\mathbf{Y}(I),r}}\Big) \cdot \\ &\Big(\sum_{T=1}^{X} \frac{\hat{p}_{\mathbf{Y}(I+1),j}(T)\Big(1 - \hat{p}_{\mathbf{Y}(I+1),j}(T)\Big)}{\sum_r F_{\mathbf{Y}(I+1),r}} + 2\Big(\sum_{T_i=1}^{X-1} \sum_{T_j=T_i}^{T_j=X} - \frac{\hat{p}_{\mathbf{Y}(I+1),j}(T_i) \cdot \hat{p}_{\mathbf{Y}(I+1),j}(T_j)}{\sum_r F_{\mathbf{Y}(I+1),r}}\Big)\Big) \end{aligned} \tag{4.25}$$

$$
\begin{aligned}
&\mathrm{Cov}(A_i, A_j) \\
&= \sum_{T_l, T_o=1}^{X} \Big(\Big(\hat{P}_{\mathbf{Y}(I+1),i} \cdot \hat{P}_{\mathbf{Y}(I+1),i} - \frac{\hat{P}_{\mathbf{Y}(I+1),i} \cdot \hat{P}_{\mathbf{Y}(I+1),j}}{\sum_r F_{\mathbf{Y}(I),r}} \Big) \cdot \hat{p}_{\mathbf{Y}(I+1),i}(T_l) \cdot \hat{p}_{\mathbf{Y}(I+1),j}(T_o) - \\
&\hat{P}_{\mathbf{Y}(I+1),i} \cdot \hat{p}_{\mathbf{Y}(I+1),i}(T_l) \cdot \hat{P}_{\mathbf{Y}(I+1),j} \cdot \hat{p}_{\mathbf{Y}(I+1),j}(T_o) \Big) - \frac{\hat{P}_{\mathbf{Y}(I+1),i} \cdot \hat{P}_{\mathbf{Y}(I+1),j}}{\sum_r F_{\mathbf{Y}(I),r}} + \\
&\sum_{T_o=1}^{X} \Big(\hat{p}_{\mathbf{Y}(I+1),j}(T_o) \cdot \Big(\hat{P}_{\mathbf{Y}(I+1),i} \cdot \hat{P}_{\mathbf{Y}(I+1),j} - \frac{\hat{P}_{\mathbf{Y}(I+1),i} \cdot \hat{P}_{\mathbf{Y}(I+1),j}}{\sum_r F_{\mathbf{Y}(I),r}} \Big) - \hat{P}_{\mathbf{Y}(I+1),i} \cdot \hat{p}_{\mathbf{Y}(I+1),j}(T_o) \cdot \hat{P}_{\mathbf{Y}(I+1),j} \Big) + \\
&\sum_{T_l=1}^{X} \Big(\hat{p}_{\mathbf{Y}(I+1),i}(T_l) \cdot \Big(\hat{P}_{\mathbf{Y}(I+1),i} \cdot \hat{P}_{\mathbf{Y}(I+1),j} - \frac{\hat{P}_{\mathbf{Y}(I+1),i} \cdot \hat{P}_{\mathbf{Y}(I+1),j}}{\sum_r F_{\mathbf{Y}(I),r}} \Big) - \hat{P}_{\mathbf{Y}(I+1),i} \cdot \hat{p}_{\mathbf{Y}(I+1),i}(T_l) \cdot \hat{P}_{\mathbf{Y}(I+1),j} \Big)
\end{aligned} \tag{4.26}
$$

$$
\mathrm{E}(A) = \sum_{i=0}^{m} \Big(1 - \sum_{T=1}^{X} \hat{p}_{\mathbf{Y}(I+1),i}(T) \Big) \cdot (\hat{P}_{\mathbf{Y}(I+1),i}) \tag{4.27}
$$

$\mathrm{Var}(B) = \mathrm{Var}\Big(\prod_{q=2}^{I} \hat{f}_{T|\mathbf{Y}(q),z}(K_1 - K_{q-1}) \cdot \hat{P}\Big(S(K_q)|z, \mathbf{Y}(q-1)\Big) \cdot \hat{P}\Big(S(K_1)|z\Big) \Big)$

Let

$C = \prod_{q=2}^{I} \hat{f}_{T|\mathbf{Y}(q),z}(K_1 - K_{q-1})$

$D = \hat{P}\Big(S(K_1)|z\Big) \cdot \prod_{w=2}^{l} \hat{P}\Big(S(K_w)|z, \mathbf{Y}(w-1)\Big)$

Then we have:

$$
\mathrm{Var}(B) = \mathrm{Var}(C \cdot D) = \mathrm{E}^2(C) \cdot \mathrm{Var}(D) + \mathrm{E}^2(D) \cdot \mathrm{Var}(C) + \mathrm{Var}(D) \cdot \mathrm{Var}(C) \tag{4.28}
$$

where:

$$
\begin{aligned}
&\mathrm{Var}(C) \\
&= \prod_{q=2}^{I} \Big(\Big(\hat{p}_{\mathbf{Y}(q),S(K_q)}(K_q - K_{q-1}) \Big)^2 + \frac{\hat{P}_{\mathbf{Y}(w),S(K_w)}(1 - \hat{P}_{\mathbf{Y}(w),S(K_w)})}{\sum_r F_{\mathbf{Y}(q),r}} \Big) - \\
&\prod_{q=2}^{I} \Big(\hat{p}_{\mathbf{Y}(q),S(K_q)}(K_q - K_{q-1}) \Big)^2
\end{aligned} \tag{4.29}
$$

$$
\begin{aligned}
&\mathrm{Var}(D) \\
&=\hat{P}^2\Big(S(K_1)|z\Big)\cdot\Big(\prod_{w=2}^{I}\Big((\hat{P}_{\mathbf{Y},S(K_w)})^2+\frac{\hat{P}_{\mathbf{Y},S(K_w)}(1-\hat{P}_{\mathbf{Y},S(K_w)})}{\sum_r F_{\mathbf{Y}(w-1),r}}\Big)-\prod_{w=2}^{I}(\hat{P}_{\mathbf{Y},S(K_w)})^2\Big)+ \\
&\Big(\prod_{w=2}^{I}\hat{P}_{\mathbf{Y},S(K_w)}\Big)^2\cdot\frac{\hat{P}\Big(S(K_1)|z\Big)\Big(1-\hat{P}\Big(S(K_1)|z\Big)\Big)}{\sum_u D_u}+\frac{\hat{P}\Big(S(K_1)|z\Big)\Big(1-\hat{P}\Big(S(K_1)|z\Big)\Big)}{\sum_u D_u}\cdot \\
&\Big(\prod_{w=2}^{I}\Big((\hat{P}_{\mathbf{Y},S(K_w)})^2+\frac{\hat{P}_{\mathbf{Y},S(K_w)}(1-\hat{P}_{\mathbf{Y},S(K_w)})}{\sum_r F_{\mathbf{Y}(w-1),r}}\Big)-\prod_{w=2}^{I}(\hat{P}_{\mathbf{Y},S(K_w)})^2\Big)
\end{aligned}
\tag{4.30}
$$

$$
\mathrm{E}(C)=\prod_{q=2}^{I}\hat{p}_{\mathbf{Y}(q),S(K_q)}(K_q-K_{q-1}) \tag{4.31}
$$

$$
\mathrm{E}(D)=\hat{P}\Big(S(K_1)|z\Big)\cdot\prod_{w=2}^{I}\hat{P}_{\mathbf{Y}(w),S(K_w)} \tag{4.32}
$$

$$
\mathrm{E}(B)=\hat{P}\Big(S(K_1)|z\Big)\cdot\Big(\prod_{q=2}^{I}\hat{P}_{\mathbf{Y}(q),S(K_q)}\cdot\hat{p}_{\mathbf{Y}(q),S(K_q)}(K_q-K_{q-1})\Big) \tag{4.33}
$$

Through Equation 4.24 to 4.33, we can get the variance of $P(\mathbf{Y}(I),X|z)$ in Equation 4.23.

5 CONTROL OF KEY PERFORMANCE INDICATORS OF MANUFACTURING PRODUCTION SYSTEMS THROUGH PAIR COPULA MODELING AND STOCHASTIC OPTIMIZATION*

Key performance indicators (KPIs) modeling and control is important for efficient design and operation of complex manufacturing production systems. This paper proposes to implement the KPI control based on KPI modeling and stochastic optimization. The KPI relationship is first approximated using ordered block model and pair-copula construction (OBM-PCC) model, which is a non-parametric model that facilitates a flexible surrogate of the KPI relationship. Then, the KPI control is framed into a stochastic optimization problem, where the randomness in the cost function depends on the decision variables. To solve this stochastic optimization problem, the standard uniform distribution is employed to link the OBM-PCC model and the cost function to degenerate the problem into an ordinary stochastic optimization problem. The proposed method is efficient in KPI control and the performance is robust to the cost function. Extensive numerical studies and comparisons are presented to demonstrate the effectiveness of the proposed KPI control framework.

5.1 Introduction

Key Performance Indicators (KPIs) are designed to measure critical system objectives and conduct performance evaluation and control of complex manufacturing systems. It is well documented that the prediction and quantification of KPIs relationship will lead to a better

*This chapter is based on the paper: **Wang C.** and Zhou S. , "Control of Key Performance Indicators of Manufacturing Production Systems through Pair Copula Modeling and Stochastic Approximation", *IISE Transactions*, under review.

understanding of the system and further benefit the system productivity improvement through KPI control [38, 40]. In practice, we often need to control the system KPI to be at certain level to optimize the operation and maximize the profit. To achieve this goal, the relationship between the KPIs and controllable variables (inputs) need to be identified and quantified. Then, a specific control problem needs to be formulated based on the relationship, and finally be solved efficiently.

To study the KPI modeling and control problem in manufacturing production systems, a stochastic queuing model is often used. For example, Buzacott and Shanthikumar [41] employed the stochastic queuing to model, predict and optimize various manufacturing systems including flow lines, transfer lines, dynamic job shops and flexible assembly systems. A typical manufacturing process can be represented by a multi-stage queue in Fig. 5.1 (a), where we have $l-1$ servers and the corresponding $l-1$ service rates, buffers and queuing time in the buffers. We represent the μ_i and q_j as the service rate and queuing time for the server i and buffer j, respectively. The $N_j \in \{1, 2, \cdots\}$ is the fixed buffer size and the λ is arrival rate. It is straightforward to see that the λ and μ_is are the inputs of the manufacturing process and the q_js are the system responses. The overall layout of the system inputs and responses is shown in Fig. 1 (b), where we define the λ and μ_is as the system input parameters and the q_js and c as the response KPIs. We denote c as process cycle time and p as total number of system input parameters and response KPIs. It is well known that for a queuing system, the response KPIs are stochastic for given fixed system input parameters even under stationary condition. As a result, the relationship between system input parameters and response KPIs is quantified by the multivariate distributions of response KPIs conditional on system input parameters, e.g., $f(q_1, q_2, c|\lambda, \mu_1)$. If these multivariate conditional distributions can be efficiently obtained and evaluated, we can have better understanding and control of the queuing networks. The M/M/1 queue, for

example, is the elementary and widely studied model in queuing theory, which facilitates the close-form expressions of such conditional distributions thus the relationship between system input parameters and many response KPIs can be explicitly characterized. The dependence structure for a M/M/1 queue is shown in Fig. 5.1 (c). Various studies were conducted to provide control guidance/policy on system input parameters $(\lambda, \mu_1, \mu_2, \cdots)$ for maximizing the KPI performance in M/M/1 queue (networks). One of the most representative and general works in the KPI control was introduced by Stidham and Weber [115], where they assumed the arrival rate is fixed and showed that under weak conditions there exists a monotone optimal policy for the service rate control problem in the M/M/1 queue. An important application and extension of [115] is the design and control of call centers, where the objective is to minimize the operation costs by selecting the staffing level for each call server and determining a routing rule for assigning calls to different severs [116, 117]. Although the above mentioned works in queuing control and analysis are of great importance and success, it is pointed out that the M/M/1 based queue can only cover the arrival and service time interval pattern in limited scenarios [118]. Meanwhile, the analytical relationship between system input parameters and response KPIs and the theoretical properties/policies of KPI control in M/M/1 queue are very difficult to be generalized to different queuing models.

To facilitate a general framework for KPIs control in manufacturing systems, we need i) an efficient surrogate model to quantify the relationship between system input parameters and response KPIs, and ii) a flexible KPI control framework that integrates the surrogate model to optimize the KPI control objective function. In the stochastic queuing context, various data driven surrogate models were proposed to accurately approximate the KPI relationship. These works can be classified into simulation based models [43, 44, 45, 46] and regression based models [47, 119, 120, 121, 122]. The simulation based models are

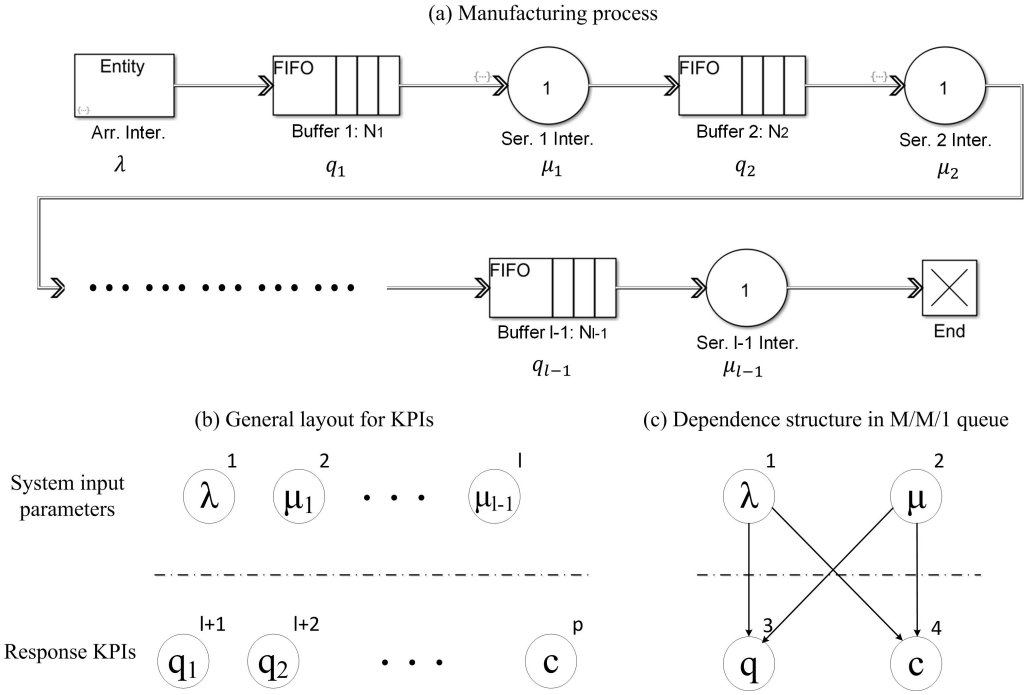


Figure 5.1: KPIs in a manufacturing process.

very general and can be applied to many complex queuing scenarios, but these models suffer from the computational issue and are difficult to adapt to customized changes in the queuing systems. The regression based models, especially the Kriging based models, are very flexible in modeling KPI relationship. However, the Kriging based surrogate models also require high computational load in learning and predicting the surrogate model, which significantly impairs the efficiency of the whole algorithm. Furthermore, these methods mainly focus on the relationship among the moments of KPIs (e.g., mean), and the comprehensive multivariate distribution information of KPIs is not provided. To characterize the multivariate distribution among KPIs and quantify the KPI relationship, we adopt the ordered block model-pair copula construction (OBM-PCC) method proposed in Chapter 3 to approximate KPI joint distribution [123] in this work. The OBM-PCC is a non-parametric method that can efficiently capture the KPI relationship in distribution

level and provide accurate surrogate samples of the system input parameters and response KPIs for various queuing systems. We will give a brief review of the OBM-PCC based surrogate model in Section II-B.

The second step in KPI control is to formulate the control problem using the surrogate model. In this chapter, we consider a serial queuing model in Fig. 5.1 (a) with general arrival and service distributions, where the holding costs and service costs are non-decreasing functions of the mean of queuing time/queuing length/cycle time and service rate, respectively. The objective is to find the system input parameters that minimize the overall cost for the queuing system in steady state. To solve this problem, a stochastic optimization framework is constructed, where the system input parameters serve as decision variables and the KPIs are in the objective functions. For example, we consider a service rate control problem in a G/G/1 queuing with fixed arrival rate: $\min_{\mu} \mathrm{E}[q + c + \mu]$. The critical challenge in this optimization setting is that the distributions of variables q and c depend on the decision variable μ, which requires repetitive evaluations of distributions of q and c in the optimization process thus impairs the optimization efficiency. Such optimization problem is called endogenous noise based optimization [124], where the 'noise' represents the uncertainty in the cost function and the 'endogenous' means the uncertainty distribution depends on the decision variables. The exogenous noise based optimization is the other type of optimization problem, where the cost function uncertainty distribution is independent with the decision variables. Please note we would use the 'endogenous noise' and 'exogenous noise' to represent these two types of optimization problems in the following descriptions. The common way to solve the endogenous noise problem is to use generative data to iteratively optimize the decision variable values through repetitive evaluations of the objective function (and its gradient) at each decision variable candidate value [125, 126]. This idea is widely used in sample average approximation, stochastic

approximation and metamodel optimization. However, it is prohibitive to use generative data in queuing model since the evaluation and data preparation at new system input parameters needs long time (run a new queuing simulation until stationary). Please note the Kriging based surrogate model for KPI cannot be used to generate data for objective function evaluation since the Kriging based surrogate model only describes the mean of each KPI instead of the mean of a function of each KPI (objective function). The decision dependent stochastic programming (DDSP) [127] is another way for solving stochastic optimization with endogenous noise. However, existing DDSP techniques [128, 129] are based on linear approximation of the objective function and can only be applied to binary decision variables, which is not applicable in the queuing context.

To solve the KPI control problem, we propose to accommodate the OBM-PCC model to the stochastic optimization framework to degenerate the original endogenous noise problem into an exogenous noise one. The basic idea is to first perform a strictly monotone transformation of both system input parameters and response KPIs. Then the joint distribution of the transformed system input parameters and response KPIs can be approximated by the pair-copula using independent and identically distributed (i.i.d.) random variables. As a result, the randomness and dependence between the transformed system input parameters and response KPIs is fully represented by the i.i.d. random variables using pair-copula, which are exogenous. We illustrate this idea using one dimension decision variable x and the corresponding objective function $G(x) = \mathrm{E}g\Big(x, \xi(x)\Big)$, where the $\xi(x)$ is the randomness in the cost function that depends on the decision variable. The proposed method enables another formulation of the objective function with independent randomness (exogenous): $\tilde{G}(\tilde{x}) = \mathrm{E}\tilde{g}\Big(\tilde{x}, \xi\Big)$, where $\tilde{x}$ is the transformed x, $\tilde{g}$ is constructed by the OBM-PCC model and the ξ is a random variable independent of $\tilde{x}$. Please note that the evaluation of the original objective function needs to run the

time-consuming simulations due to the endogenous $\xi(x)$, whereas the evaluation on the transformed model is efficient. With the transformed model, we can perform standard stochastic approximation methods to efficiently solve the optimization problem in the transformed domain. The last step in our method is to show the objective function equivalence between the original problem and the transformed problem. To solve this issue, we propose a lemma to demonstrate that the exogenous property in the transformed system input parameters and response KPIs can be inherited to the original problem so that the optimization of the transformed problem is equivalent to solving the original problem. The key feature in the proposed method is that i) the pair-copula formulation enables the i.i.d. random variables to quantify the relationship among the transformed variables and ii) we discover the equivalent objective function that employs the pair-copula and the i.i.d. standard uniform random variables to degenerate the original objective function into an exogenous one. To solve the degenerated problem, we resort to simultaneous perturbation stochastic approximation (SPSA) to obtain the optimized system input parameters for KPI control.

The rest of the chapter is organized as follows: Section 5.2 gives the detailed problem formulation and presents the proposed OBM-PCC based stochastic optimization for KPI control. Numerical experiments will be conducted in Section 5.3 to show the effectiveness of the proposed method. Section 5.4 draws conclusion remarks.

5.2 problem formulation and KPI control

5.2.1 Problem formulation and assumptions

In this work, we model the production system as a queuing system. We will use the same assumptions and notations as those in Chapter 3, and omit the elaborations here.

To mathematically study the KPI control problem, we formulate the objective function and constraints as follows:

$$\begin{aligned} &\min_{\boldsymbol{x}_I} \mathrm{E}_{\boldsymbol{x}_O|\boldsymbol{x}_I}\Big[g(\boldsymbol{x}_I, \boldsymbol{x}_O)\Big] \\ &\text{subject to: } B_-(i) \leq x_i \leq B_+(i), i = 1, \cdots, l, \end{aligned} \tag{5.1}$$

where $\boldsymbol{x}_I = (x_1, \cdots, x_l)$ is the system input parameter variables vector (decision variables), $\boldsymbol{x}_O = (x_{l+1}, \cdots, x_p)$ is the KPI variables vector, $B_-(i)$ and $B_+(i)$ are real values that represent the lower and upper constraints for $x_i, i = 1, \cdots, l$. The function $g(\cdot)$ is a cost function that applies to $\boldsymbol{x}_I$ and $\boldsymbol{x}_O$, and the conditional expectation of $g(\boldsymbol{x}_I, \boldsymbol{x}_O)$ with respect to $\boldsymbol{x}_O|\boldsymbol{x}_I$ constructs the objective function.

Equation 5.1 is a general form for stochastic optimization with endogenous variables in the cost function. The stochastic optimization with endogenous noise is in general very difficult to solve, but there are unique features in parameters relationship in queuing models that allow us to solve the problem numerically using PCC technique. We will first review the OBM-PCC model in section 5.2.2 to provide a surrogate model of KPI relationship. Then, the details of solving Equation 5.1 will be introduced in section 5.2.3.

5.2.2 Review of OBM-PCC model

We adopt the method proposed in Chapter 3 to approximate the KPI joint distribution to numerically represent the KPI relationship [123]. The copula itself is a multivariate cumulative distribution function (cdf) $C : [0, 1]^p \to [0, 1], p \in \mathbb{N}$ such that all the univariate marginals are uniform distributions on the interval $[0, 1]$ [54]. The Sklar's theorem [61] shows for an arbitrary continuous joint probability distribution, we have:

$$f(x_1, ..., x_p) = c\Big(F_1(x_1), ..., F_p(x_p)\Big) \prod_{i=1}^{p} f_i(x_i), \tag{5.2}$$

where $F_i(\cdot)$ is the marginal cdf for variable x_i, and the copula pdf c is uniquely determined. The Sklar's theorem lays the theoretical foundation for joint distribution fitting using copula: With the result in Equation 5.2, we can estimate the marginal distribution for each variable and then estimate the joint copula function $c(\cdot)$. However, the direct estimation of high dimensional copula function $c(\cdot)$ is challenging. To simply the fitting process, the PCC was proposed to decompose the high dimensional copula fitting into multiple pair-wise (bi-variate) copula fittings [55, 64]. Moreover, the special dependence structure between system input parameters and response KPIs in queuing models can provide additional information for PCC. For example, the dependence structure of system input parameters and response KPIs in the M/M/1 queuing is shown in Fig. 5.1 (c), where only the system input parameters can have influence on the response KPIs. The relationship in Fig. 5.1 (c) gives a graphical representation of the dependence structure, which can be formally described as an OBM [5]. The basic idea in OBM-PCC model is to use the OBM to capture the qualitative dependence information between system input parameters and response KPIs, then the dependence information is integrated into the PCC to facilitate a more efficient and accurate approximation of the joint distribution among queuing variables. As a result, the high dimensional copula can be simplified using OBM-PCC model as follows:

$$c\Big(F_1(x_1),...,F_p(x_p)\Big)=\prod_{i=1}^{p}\prod_{w\in pa(i)}c_{iw|pa(i;w)}\Big(F_{i|pa(i;w)}(x_i|\boldsymbol{x}_{pa(i;w)}),F_{w|pa(i;w)}(x_w|\boldsymbol{x}_{pa(i;w)})\Big), \tag{5.3}$$

where $\boldsymbol{x}_{pa(i)}$ is the set of parent variables of X_i, e.g., $\boldsymbol{X}_{pa(3)}=\{X_1,X_2\}$ in Fig. 5.1 (c), $pa(i;w)$ is the wth set of parent variables of node i, $c_{iw|pa(i;w)}$ is a conditional pair-copula describing the variable dependence between i and w conditional on $pa(i;w)$. The

conditional cdfs $F_{j|K}, j \in V, K \subseteq V \backslash j$ can be computed using the recursive formula:

$$F_{j|K}(x_j|\mathbf{x}_K) = \frac{\partial C_{jk|(K\backslash\{k\})}\Big(F_{j|(K\backslash\{k\})}(x_j|\mathbf{x}_{(K\backslash\{k\})}), F_{k|(K\backslash\{k\})}(x_k|\mathbf{x}_{(K\backslash\{k\})})\Big)}{\partial F_{k|(K\backslash\{k\})}(x_k|\mathbf{x}_{(K\backslash\{k\})})}, \tag{5.4}$$

for $k \in K$.

The parameter estimation and copula type selection are conducted through maximum likelihood estimation and Akaike information criterion [69]. The estimated OBM-PCC model is a surrogate model for $f(x_1, \cdots, x_p)$, and large number of samples can be efficiently generated based on the surrogate model to represent the stochastic relationship among p variables. More details about the OBM-PCC model can be referred to Chapter 3.

5.2.3 KPI control

The KPI control problem in Equation 5.1 is in general difficult to solve directly since the endogenous noise changes with the decision variables. We propose to employ the OBM-PCC model to separate the endogenous noise from decision variable and degenerate the problem in Equation 5.1 into a traditional stochastic optimization. To achieve this goal, we need to perform a strictly monotone transformation T_i on x_i to generate the standard uniformly distributed variables that facilitate the OBM-PCC model:

$$\tilde{x}_i = T_i(x_i), i = 1, \cdots, p \tag{5.5}$$

where the T_i can be kernel transformations or empirical cdfs. We further denote $\tilde{\boldsymbol{x}}_I = (\tilde{x}_1, \cdots, \tilde{x}_l)$. With the Equation 5.4 and the OBM-PCC model, we can get the transformed

conditional response KPIs:

$$\tilde{x}_i|\tilde{\boldsymbol{x}}_I = h_i(\tilde{\boldsymbol{x}}_I, \xi_i), i = l+1, \cdots, p \tag{5.6}$$

where

$$h_i(\tilde{\boldsymbol{x}}_I, \xi_i) = \begin{cases} F^{-1}_{l+1|1,\ldots,l}(\xi_{l+1}|\tilde{\boldsymbol{x}}_I; \boldsymbol{\theta}) & i = l+1 \\ F^{-1}_{i|1,\ldots,i-1}(\xi_i|\tilde{\boldsymbol{x}}_I, \tilde{x}_{l+1}, ..., \tilde{x}_{i-1}; \boldsymbol{\theta}) & i = l+2, \cdots, p \end{cases} \tag{5.7}$$

is defined by the inverse function of Equation 5.4 with $\boldsymbol{\theta}$ as the estimated OBM-PCC parameters and ξ_i as the i.i.d standard uniform distributed variables.

With Equation 5.5 and 5.7, we can degenerate the problem in Equation 5.1 into an exogenous noise based optimization, and we have the lemma as follows:

Lemma 3: Given the $T_i(\cdot)$ is a strictly monotone transformation and the $h_i(\cdot, \xi_i)$ is a strictly monotone function on ξ_i, Equation 5.1 can be degenerated with the OBM-PCC model as follows:

$$\begin{gathered} \min_{\tilde{\boldsymbol{x}}_I} \mathrm{E}_{\boldsymbol{\xi}}\left[g\Big(T_1^{-1}(\tilde{x}_1), \cdots, T_l^{-1}(\tilde{x}_l), T_{l+1}^{-1}\Big(h_{l+1}(\tilde{\boldsymbol{x}}_I, \xi_{l+1})\Big), \cdots, T_p^{-1}\Big(h_p(\tilde{\boldsymbol{x}}_I, \xi_p)\Big)\Big)\right] \\ \text{subject to: } T_i\Big(B_-(i)\Big) \le \tilde{x}_i \le T_i\Big(B_+(i)\Big), i = 1, \cdots, l \end{gathered} \tag{5.8}$$

where $\boldsymbol{\xi} = \{\xi_{l+1}, \cdots, \xi_p\}$. After we get the optimized $\tilde{\boldsymbol{x}}_I$, we can apply $T_i^{-1}(\cdot)$ on $\tilde{\boldsymbol{x}}_I$ to get the $\boldsymbol{x}_I$. Please note that in Equation 5.8, the $\{T_1^{-1}(\tilde{x}_1), \cdots, T_l^{-1}(\tilde{x}_l)\}$ is $\boldsymbol{x}_I$ and the $\{T_{l+1}^{-1}\Big(h_{l+1}(\tilde{\boldsymbol{x}}_I, \xi_{l+1})\Big), \cdots, T_p^{-1}\Big(h_p(\tilde{\boldsymbol{x}}_I, \xi_p)\Big)\}$ is $\boldsymbol{x}_O$. Thus we only re-formulate the problem with transformed decision variables and do not change the objective function. The derivation of the lemma is in the appendix.

Equation 5.8 has three notable features that simplify the KPI control problem.

i Equation 5.8 becomes a stochastic optimization with exogenous noise. The noise in the cost function is defined by the i.i.d. standard uniform

variable vector $\boldsymbol{\xi}$, and the decision variables $\tilde{\boldsymbol{x}}_I$ have no influence on the noise. The reason Equation 5.8 becomes exogenous is that the function $h_i(\tilde{\boldsymbol{x}}_I, \xi_i)$ can represent the dependence between $\tilde{\boldsymbol{x}}_I$ and each transformed response KPI variable with exogenous noise ξ_i in an explicit form using pair-copula construction. We can indeed represent such relationship using other tools, e.g., Kriging or spline, but the final objective function may not become exogenous. This important feature degenerates the Equation 5.1 into an ordinary stochastic optimization problem.

ii The efficient evaluation of the cost function g. For a specific $\tilde{\boldsymbol{x}}_I$ and $\boldsymbol{\xi}$, the function h_i typically has a close-form expression determined by the selected pair-copula type and estimated parameters. The inverse function T_i^{-1} might not have an explicit form, but it is a one-to-one mapping that is efficient to evaluate. We do not need to re-run the queuing model with new set of $\tilde{\boldsymbol{x}}_I$ during the optimization, which significantly improve the algorithm efficiency.

iii Adjustable uncertainty level in the cost function. The cost function uncertainty adjustability is important for convergence efficiency in stochastic approximation methods [130]. The uncertainty in the cost function depends on the form of the cost function g and how we evaluate the g. In solving the Equation 5.8 using stochastic approximation, we can generate several $\boldsymbol{\xi}$ vectors and take the average over sampled $g(\cdot, \boldsymbol{\xi})$ as the cost function to adjust the uncertainty level. This is an efficient process due to the property in ii), while the same process for endogenous problems would involve extra computation load in evaluating the distribution of cost functions.

We want to point out that the optimization performance of Equation 5.8 also depends on the OBM-PCC accuracy in approximating the queuing models. The intuition the OBM-PCC model can accurately represent the general queuing model is that the pair-copula is invariant under monotone transformation [131], which means all the pair-copula types and parameters remain the same if we apply a monotone transformation to each of $(x_1, \cdots, x_p)$. This property provides great convenience in validating the OBM-PCC model through trying different transformation functions $T_i(\cdot)$. In practice, we can first validate the accuracy of the fitted OBM-PCC model through Kolmogorov-Smirnov test [72], then apply the validated model into Equation 5.8. As a result, these three features together contribute to the efficiency of the KPI control problem in queuing models.

To solve Equation 5.8, we resort to simultaneous perturbation stochastic approximation method that specifically addresses multivariate optimization problems [132]. The SPSA only requires the objective function values to approximate the underlying gradient and is therefore easy to implement. Moreover, SPSA only requires two function evaluations at each iteration regardless of the dimension of the decision variable space, which could potentially reduce the computational cost significantly in high dimensional problems. The general idea of SPSA is as follows:

$$\begin{aligned} \tilde{\boldsymbol{x}}_{I,n+1} &= \prod_{\Theta} \left(\tilde{\boldsymbol{x}}_{I,n} - a_n \hat{\nabla} g(\tilde{\boldsymbol{x}}_{I,n}) \right) \\ \hat{\nabla} g_i(\tilde{\boldsymbol{x}}_{I,n}) &= \frac{g(\tilde{\boldsymbol{x}}_{I,n} + d_n \Delta_n, \xi_n^+) - g(\tilde{\boldsymbol{x}}_{I,n} - d_n \Delta_n, \xi_n^-)}{2 d_n \Delta_{n,i}} \end{aligned} \tag{5.9}$$

where n is the iteration index, $\prod_{\Theta}(\tilde{\boldsymbol{x}}_I)$ is a projection of $\tilde{\boldsymbol{x}}_I$ back into the feasible region Θ defined by the constraints [**?**], $g(\cdot, \xi)$ is the representation of objective function of Equation 5.8, $\hat{\nabla} g_i(\tilde{\boldsymbol{x}}_{I,n})$ is the estimate of the ith entry of the gradient $\nabla g(\tilde{\boldsymbol{x}}_{I,n})$, $a_n > 0$ is the step size, $\Delta_n = (\Delta_{n,1}, \cdots, \Delta_{n,1}) \in \mathbb{R}^p$ is assumed to be i.i.d. distributed, $d_n > 0$ is the

finite difference step size, and $\xi_n^{\pm}$ denotes the randomness. The optimal convergence rate for SPSA is $\mathcal{O}(n^{-1/3})$. The perturbation sequence $\Delta_n = (\Delta_{n,1}, \cdots, \Delta_{n,1})$ must have mean zero and finite inverse moments. As a result, the Gaussian distribution is not applicable. One of the most common distributions for $\{\Delta_{n,i}\}$ is the symmetric Bernoulli taking a positive and negative value, i.e., ± 1 with probability 0.5 [130], and we will use this setting in the simulation studies. In addition, an appropriately scaled $\tilde{\boldsymbol{x}}_{I,n}$ is asymptotically normal for large n, and the relative efficiency of SPSA also depends on the choice of $\{a_n\}$, $\{d_n\}$ and the noise level.

We summarize the algorithm for KPI control in algorithm 4.

Algorithm 4 KPI control

Input: Sample vectors for each variable $\boldsymbol{x}_1, ..., \boldsymbol{x}_p$;

The system input parameter and response KPI variables sets **I** and **O**;

The cost function $g(\boldsymbol{x}_I, \boldsymbol{x}_O)$.

1. Transform the sample vector $\boldsymbol{x}_i$ to uniformly distributed using Equation 5.5;
2. Fit the OBM-PCC model and validate the accuracy;
3. Re-formulate the objective function using Equation 5.8;
4. Choose the appropriate $\{a_n\}$ and $\{d_n\}$ for SPSA;
5. Solve the Equation 5.8 using SPSA through Equation 5.9 and obtain the optimal solution $\tilde{\boldsymbol{x}}_{I,opt}$;
5. Apply $T_i^{-1}(\cdot)$ to $\tilde{\boldsymbol{x}}_{I,opt}$ to get the desirable KPI control input $\boldsymbol{x}_{I,opt}$.

Output: $\boldsymbol{x}_{I,opt}$.

5.3 Numerical studies

Various numerical studies are implemented in this section to validate the effective of the proposed OBM-PCC based KPI control strategy. We provide three settings using different queuing models: i) M/M/1 queuing network with two servers and infinite buffer size; ii)

G/G/1 queuing network with two servers and infinite buffer size; iii) G/G/1 queuing network with four servers and finite buffer size. We first apply the OBM-PCC model to each of the queuing model to build the surrogate model, then the KPI control will be conducted for different cost functions. The KPI control quality will be evaluated using the relative error of the optimization results. To demonstrate the superiority of proposed method, we compare the method with the Kriging based KPI control. The basic idea in Kriging based optimization is to treat the objective function as a unknown function of decision variables, and this unknown function will be modeled/predicted using noisy readings at available decision variables. More specifically, the Gaussian process [133] and stochastic Kriging [122] will be used to model the response and noise of the objective function. For example, the objective function in Equation 5.1 will be re-formulated for Kriging based optimization as follows:

$$\begin{gathered} \min_{\boldsymbol{x}_I} g^*(\boldsymbol{x}_I) \\ \text{subject to: } B_-(i) \leq x_i \leq B_+(i), i = 1, \cdots, l \end{gathered} \tag{5.10}$$

where $g^*(\boldsymbol{x}_I) = \mathrm{E}_{\boldsymbol{x}_O|\boldsymbol{x}_I}\Big[g(\boldsymbol{x}_I, \boldsymbol{x}_O)\Big]$. Based on the available samples of $\boldsymbol{x}_I$, the queuing simulation could provide samples of $\boldsymbol{x}_O$ conditional on $\boldsymbol{x}_I$, and the samples of $g^*(\boldsymbol{x}_I)$ can be obtained consequently. The Gaussian process or the stochastic Kriging will then be employed to model the sample tuples $\Big(\boldsymbol{x}_I, g^*(\boldsymbol{x}_I)\Big)$ to interpolate the function $g^*(\cdot)$ within the parameter space of $\boldsymbol{x}_I$. As a results, Equation 5.10 becomes a conventional non-linear optimization and various solvers can be applied to solve it. The main difference between the Gaussian process and stochastic Kriging is the Gaussian process assumes the noise on samples of $g^*(\boldsymbol{x}_I)$ is i.i.d. across the parameter space of $\boldsymbol{x}_I$, whereas the stochastic Kriging allows heterogeneous noise, e.g., the sample noise on $g^*(\boldsymbol{x}_I)$ depends on the value of $\boldsymbol{x}_I$. The stochastic Kriging is clearly a better choice for modeling the $g^*(\cdot)$ due to

the endogenous noise in the queuing context. However, the stochastic Kriging requires multiple samples of $g^*(\boldsymbol{x}_I)$ given a specific $\boldsymbol{x}_I$ to provide the co-variance estimation, which would consume a large number of simulation repetitions. Meanwhile, the computation efficiency is a common issue in all the Kriging based methods. We will discuss the KPI control quality in terms of computation efficiency and accuracy among different methods.

5.3.1 M/M/1 queuing network

First, we consider the M/M/1 queuing network with the first come first serve (FCFS) dispatching rule for waiting lines. The M/M/1 queue has the close form steady state distributions of queue waiting time and cycle time, and we will use these properties to identify the underlying truth of the KPI control problem for validation purpose. Figure 5.2 (a) gives the diagram of the M/M/1 queuing network under study, where the buffer sizes are set to be infinite. The simulation parameters are listed in Table 5.1. In the simulation, we set the arrival rate as fixed (λ=1) and randomly pick a pair of μ_1 and μ_2 according to the uniform distributions. Then, the service time distributions are set based on Table 5.1 for the queuing network. As the simulation reaches the steady state, we can sample and record q_1, q_2 and c. Then, we re-pick another pair of μ_1 and μ_2 and repeat the process. We perform the simulation with Simulink, Matlab 2017. We can then employ the OBM-PCC method to approximate the relationship between these variables, and the results is shown in Fig. 5.2 (b). The selected pair-copula and the parameters are attached on the lines. Based on the OBM-PCC model, we use algorithm 4 to implement the KPI control with the objective function as follows:

$$\begin{gathered} \mathrm{E}_{\boldsymbol{x}_O|\boldsymbol{x}_I}\Big[g(\boldsymbol{x}_I, \boldsymbol{x}_O)\Big] = \mathrm{E}(q_1 + q_2 + c) + 0.4\mu_1 + 0.3\mu_2 \\ \text{subject to: } 2 \leq \mu_1 \leq 4, 2 \leq \mu_2 \leq 4 \end{gathered} \tag{5.11}$$

where we set the cost rate for queuing time and cycle as 1 and the cost rate for server 1 as 0.4 and the cost rate for server 2 as 0.3. Equation 5.11 is with endogenous noise and needs to be transformed into the exogenous form in Equation 5.8 through the fitted OBM-PCC model in Fig. 5.2 (b). Then, the SPSA can be applied to solve the KPI control problem. When implementing the SPSA method, we choose the $\{a_n\}$ and $\{d_n\}$ based on the suggestions in [130] as follows:

$$\begin{aligned} a_n &= \frac{a}{(A+n+1)^{0.602}} \\ d_n &= \frac{d}{(n+1)^{0.101}} \end{aligned} \tag{5.12}$$

where a and A are related with the maximum number of iteration and the smallest desired change magnitude of $\tilde{\boldsymbol{x}}_I$ in each iteration, the larger the allowed iteration number and the smallest change magnitude, the larger the A and a will be, the d is positively related with the noise level of the cost function. In the our simulation settings, we choose $a = 0.064$, $A = 1000$ and $d = 0.8$.

Table 5.1: Parameter setting for M/M/1 queuing network.

Parameter	Type	Mean	Variance	Min	Max
Arrival interval	Exponential	1	1	0	∞
Server 1 interval	Exponential	$\frac{1}{\mu_1}$	$\frac{1}{\mu_1^2}$	0	∞
Server 2 interval	Exponential	$\frac{1}{\mu_1}$	$\frac{1}{\mu_1^2}$	0	∞
μ_1	Uniform	3	$\frac{1}{3}$	2	4
μ_2	Uniform	3	$\frac{1}{3}$	2	4

Since it is a M/M/1 based queuing network and the buffer size is infinite, we can easily get the close form of Equation 5.11 as follows:

$$\begin{aligned} \mathrm{E}_{\boldsymbol{x}_O|\boldsymbol{x}_I}\Big[g(\boldsymbol{x}_I, \boldsymbol{x}_O)\Big] &= \frac{2}{\mu_1 - 1} + \frac{2}{\mu_2 - 1} + 0.4\mu_1 + 0.3\mu_2 \\ \text{subject to: } & 2 \le \mu_1 \le 4, 2 \le \mu_2 \le 4 \end{aligned} \tag{5.13}$$

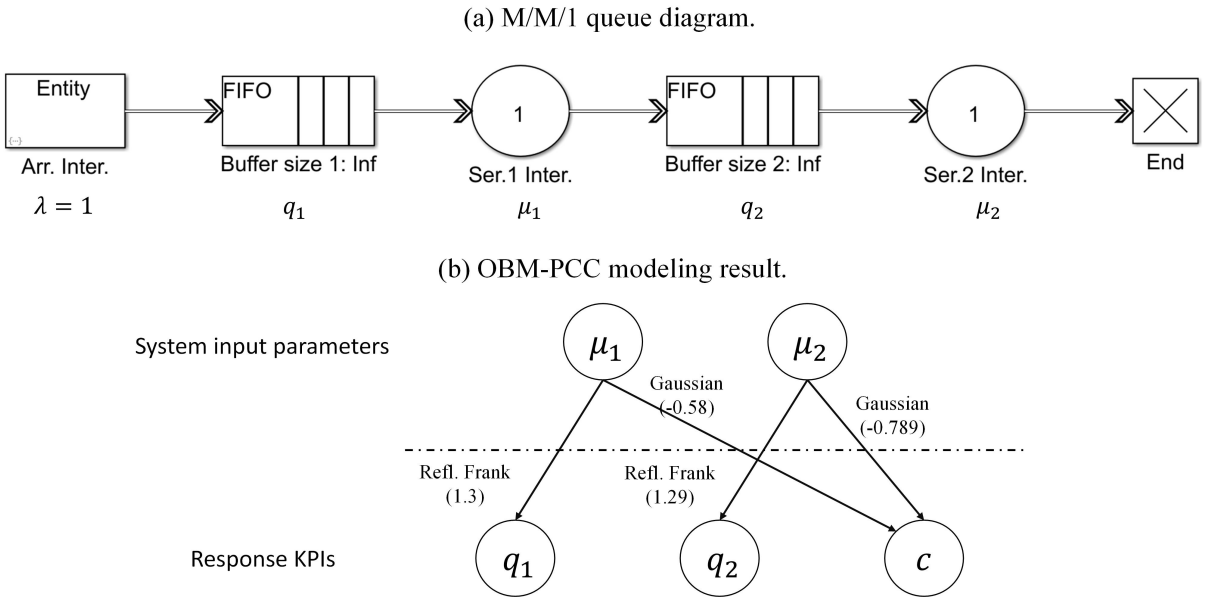


Figure 5.2: Simulation for M/M/1 queuing network.

where we only consider the scenario when queuing time is large than 0 (otherwise the conditional distribution of q would not be exactly continuous). We can then get the KPI control input parameter by minimizing Equation 5.13, which results in the minimized objective function value 4.038 when $\mu_1 = 3.236$ and $\mu_2 = 3.582$. The value 4.038 will be used as the underlying truth to compare with the optimization results of the proposed method and the Kriging based methods. The results are shown in Table 5.2, where we demonstrate the computation time and results accuracy (relative error with 4.038) among different methods under different number of $\boldsymbol{x}_I$ samples. The computation time in Table 5.2 includes the model fitting and optimization solving. For Kriging based methods, we use the Nelder-Mead simplex approach [134] to find the minimum of the predicted response surface.

In Table 5.2, we can see the OBM-PCC based KPI control performs the best with the least time consumption. The stochastic Kriging modeling demonstrates comparable

Table 5.2: KPI control results for M/M/1 queuing network.

Sample number	100	400	900
Proposed method error	16.81%	9.12%	5.76%
Time consumed	17.12s	60.83s	139.11s
Gaussian process method error	29.18%	24.39%	19.09%
Time consumed	17.89s	471s	4508s
Stochastic Kriging method error	19.45%	12.91%	8.68%
Time consumed	30.39s	907s	8876s

accuracy to that of the proposed method, but it requires much more time to achieve the satisfactory result, especially when the number of data points becomes large. This is because the stochastic Kriging needs to conduct the inverse of large matrices (dimension determined by the number of data) two times, which consumes large amount of computation resource as the matrix dimension increases. The Gaussian process based method is also time-consuming and performs the worst, which is mainly due to the lack of heterogeneous noise characterization in the Gaussian process. Another reason the Gaussian process is not suitable for KPI control is that the objective samples are very noisy. We can see the true objective function (Equation 5.13) in Fig. 5.3 (a), which is straightforward to locate the minimum point at $(\mu_1 = 3.236, \mu_2 = 3.582)$. However, the samples of objective function from the queuing simulations, shown in Fig. 5.3 (b), are too noisy to capture the profile of the objective function. This is because the cost function in Equation 5.11 has three variables and the sample variance of the Gaussian process is the addition of these three variables. On the other hand, the proposed method always deals with bi-variate (pair-copula) variance regardless of the form of the cost function. As a result, the performance of the proposed method is expected to be more robust to the cost function form and it is not surprising the Gaussian process fails in the KPI control in this setting.

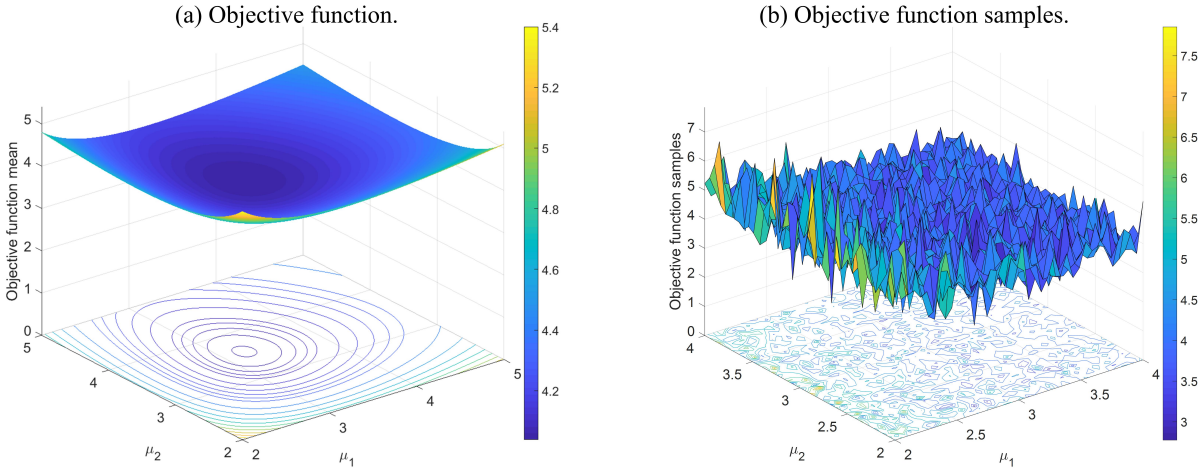


Figure 5.3: Objective function response comparison for the M/M/1 queuing network.

5.3.2 G/G/1 queuing network

The G/G/1 queuing models provide more flexibility in characterizing real systems. However, there is no general formula to capture the stationary distribution of KPIs in G/G/1 queue. As a result, the data-driven methods become compelling in G/G/1 queuing analysis. We compare the proposed method and the two Kriging based methods in G/G/1 queuing system KPI control. The queuing system configuration is the same as that in Fig. 5.2 (a) and the parameter sets are shown in Table 5.3, where the arrival interval follows log-normal distribution and the service interval follows Erlang distribution. Similar to the M/M/1 case, we run the queuing simulation in various system input parameters settings and collect the samples of KPIs. The OBM-PCC model for G/G/1 queuing network is shown in Fig. 5.4. We use a different objective function as follows:

$$\begin{aligned} \mathrm{E}_{\boldsymbol{x}_O|\boldsymbol{x}_I}\Big[g(\boldsymbol{x}_I,\boldsymbol{x}_O)\Big] &= \mathrm{E}(c) + 0.4\mu_1 + 0.3\mu_2 \\ \text{subject to: } & 2 \le \mu_1 \le 4, 2 \le \mu_2 \le 4 \end{aligned} \tag{5.14}$$

where we only consider the cycle time cost in the objective function. This may potentially reduce the observation noise for Kriging based method since the variance of cost function only comes from c. The KPI control results are shown in Table 5.4, where we run extensive simulations to figure out the objective truth for error comparisons.

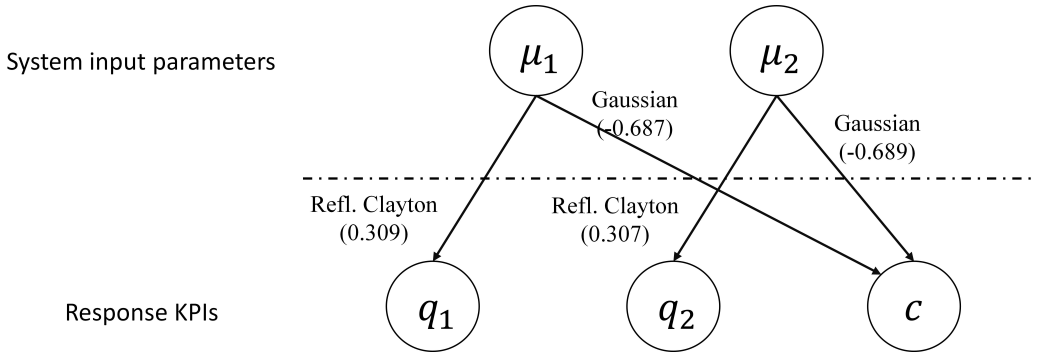


Figure 5.4: Dependence structure in the G/G/1 queuing network.

Table 5.3: Parameter setting for G/G/1 queuing network.

Parameter	Type	Mean	Variance	Min	Max
Arrival interval	Log-normal	1	1	0	∞
Server 1 interval	Erlang	$\frac{1}{\mu_1}$	$\frac{1}{2\mu_1^2}$	0	∞
Server 2 interval	Erlang	$\frac{1}{\mu_2}$	$\frac{1}{2\mu_2^2}$	0	∞
μ_1	Uniform	3	$\frac{1}{3}$	2	4
μ_2	Uniform	3	$\frac{1}{3}$	2	4

Table 5.4: KPI control results for G/G/1 queuing network.

Sample number	100	400	900
Proposed method error	16.36%	8.44%	4.91%
Time consumed	17.08s	57.94s	126.92s
Gaussian process method error	20.53%	16.00%	10.56%
Time consumed	15.48s	442s	4426s
Stochastic Kriging method error	16.45%	9.07%	4.87%
Time consumed	30.42s	857s	8517s

In Table 5.4 we can see the time consumption in three methods does not change significantly. However, there is some improvement in the KPI control result using Kriging methods, especially the Gaussian process. This observation validates the KPI control performance of the Kriging methods is sensitive to the cost function g. In Equation 5.14, the sample noise only comes from the variable c, which is much smaller than that in Equation 5.11. On the other hand, the OBM-PCC based KPI control is robust against the cost function and the performance is consistently more efficient than the Kriging based methods.

5.3.3 Serial production line

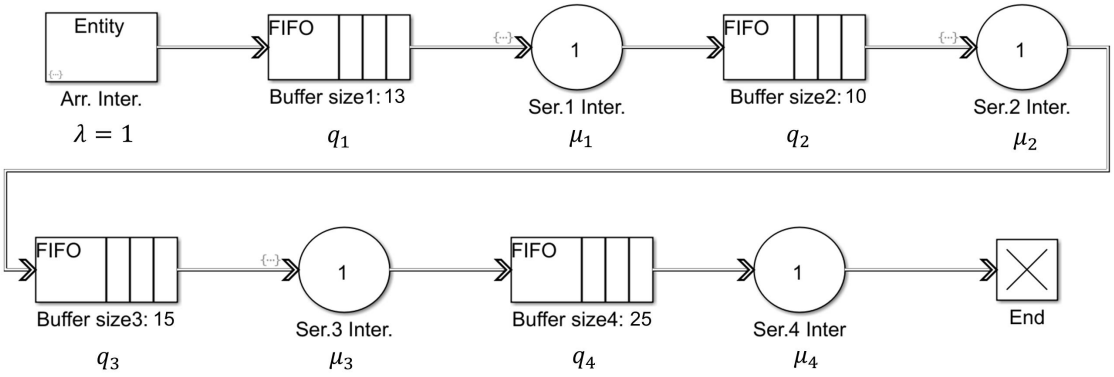
In this section, we test the KPI control performance in a serial production system with four servers. The production line diagram is shown in Fig. 5.5 (a) and (b), where the system input parameters are listed in Table 5.5 and the finite buffer sizes are attached below the servers. We consider a different objective function as follows:

$$\begin{aligned} \mathrm{E}_{\boldsymbol{x}_O|\boldsymbol{x}_I}\Big[g(\boldsymbol{x}_I,\boldsymbol{x}_O)\Big] &= \mathrm{E}(q_1q_2+q_3+q_4+c)+0.12\mu_1+0.24\mu_2+0.38\mu_3+0.31\mu_4 \\ \text{subject to: } & 4\le\mu_1\le 6, 3.5\le\mu_2\le 5.5, 3\le\mu_3\le 5, 3\le\mu_4\le 4 \end{aligned} \tag{5.15}$$

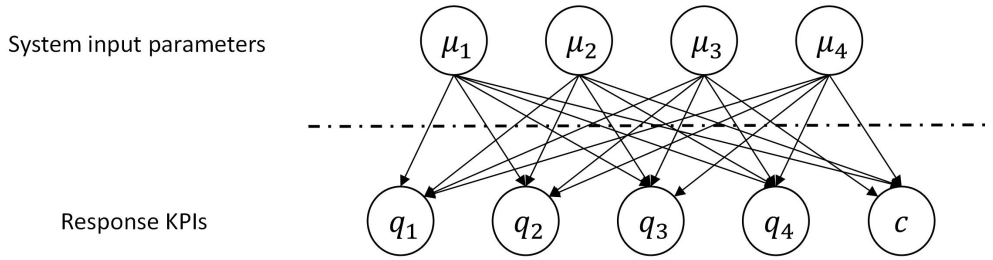
where the cost function involves quadratic terms. We demonstrate the KPI control results in Table 5.6, where we can see the OBM-PCC method again performs the best with the least time consumption.

From the extensive case studies, we find the proposed OBM-PCC based KPI control method can adapt to various production processes and achieve satisfactory control results. Please note that although the traditional queuing simulation can be directly applied to the stochastic optimization of the objective function, it would be extremely time-consuming to run the new simulation in each optimization iteration. Our method provides an effective

(a) Serial production line diagram.



(b) OBM-PCC modeling result.



$\mu_1 \to q_1$: Relf. Clayton (0.13), $\mu_1 \to q_2$: Refl. Frank (0.196), $\mu_1 \to q_3$: Refl. Joe (1.010), $\mu_1 \to q_4$: Refl. Frank (0.021), $\mu_1 \to c$: Gaussian (-0.369)

$\mu_2 \to q_1$: Relf. Clayton (0.015), $\mu_2 \to q_2$: Refl. Joe (1.35), $\mu_2 \to q_3$: Refl. Frank (0.039), $\mu_2 \to q_4$: Refl. Joe (1.040), $\mu_2 \to c$: Gaussian (-0.375)

$\mu_3 \to q_1$: Relf. Clayton (0.012), $\mu_3 \to q_2$: Refl. Frank (0.024), $\mu_3 \to q_3$: Refl. Clayton (0.225), $\mu_3 \to q_4$: Refl. Clayton (0.055), $\mu_3 \to c$: Refl. Frank (8.6)

$\mu_4 \to q_1$: Relf. Frank (0.033), $\mu_4 \to q_2$: Refl. Clayton (0.017), $\mu_4 \to q_3$: Student t (0.012, 30), $\mu_4 \to q_4$: Refl. Clayton (0.168), $\mu_4 \to c$: Gaussian (-0.396)

Figure 5.5: Simulation for serial production line.

Table 5.5: Parameters for serial production line.

Parameter	Type	Mean	Variance	Min	Max
Arrival interval	Log-normal	1	1	0	∞
Server 1 interval	Erlang	$\frac{1}{\mu_1}$	$\frac{1}{2\mu_1^2}$	0	∞
Server 2 interval	Erlang	$\frac{1}{\mu_2}$	$\frac{1}{2\mu_2^2}$	0	∞
Server 3 interval	Erlang	$\frac{1}{\mu_3}$	$\frac{1}{2\mu_3^2}$	0	∞
Server 4 interval	Erlang	$\frac{1}{\mu_4}$	$\frac{1}{2\mu_4^2}$	0	∞
μ_1	Uniform	5	$\frac{1}{3}$	4	6
μ_2	Uniform	4.5	$\frac{1}{3}$	3.5	5.5
μ_3	Uniform	4	$\frac{1}{3}$	3	5
μ_4	Uniform	3.5	$\frac{1}{12}$	3	4

Table 5.6: KPI control results for serial production line.

Sample number	100	400	900
Proposed method error	22.94%	14.17%	10.03%
Time consumed	82.7s	276.1s	541.8s
Gaussian process method error	38.53%	27.00%	21.56%
Time consumed	94.6s	1904s	16808s
Stochastic Kriging method error	26.33%	18.57%	13.40%
Time consumed	179.6s	3479s	30157s

surrogate of the queuing simulation under any system inputs, which is then integrated into the stochastic optimization to degenerate the endogenous cost function into the exogenous one. This is a very important step and contributes to the efficiency of the whole KPI control algorithm.

To guarantee the model performance, we give several important guidelines for simulation data generation and prediction procedures:

1 It is suggested to generate system input parameters samples from uniform distributions. This would simplify the copula estimation process.

2 The simulated queuing system should be able to reach the steady state, e.g., $\lambda < \mu$, and the response KPIs should be sampled in the steady state.

3 The support of uniform distributions of system input parameters should cover the constraints in the SPSA to guarantee the optimization is conducted within the OBM-PCC approximation region.

4 The sequence $\{a_n\}$ and $\{d_n\}$ needs to be chosen carefully. More details about setting the sequence parameter can be found in [130].

5.4 Conclusion

In this chapter, we propose to integrate the OBM-PCC model into the KPI control in general queuing systems. The KPI control is essentially an endogenous stochastic optimization problem, where the distribution of randomness in the cost function depends on the decision variables. The OBM-PCC model is a flexible surrogate model for approximating the system input parameter and response KPI relationship in queuing systems. We investigate the properties of this relationship and develop a KPI control scheme that assimilates the unique feature of the OBM-PCC model, which degenerates the endogenous stochastic optimization into the ordinary optimization problem. The key contribution of the chapter is that we employ the standard uniform distribution to link the OBM-PCC model and the KPI control problem so that the endogenous stochastic optimization in KPI control can be solved using existing solvers. Moreover, the developed method is validated with extensive simulation studies to show the robustness towards the cost function. The proposed method is compared with Kriging based methods in KPI control, and the results demonstrate the effectiveness and superiority of our method.

Some interesting open problems remain in the proposed method. First, the pair-copula utilized in the OBM-PCC model can only approximate monotonic relationship between variables. Although the queuing system particularly fits this requirement, it limits the application of the proposed method in general KPI control problems. Some non-monotone pair-copula can be considered to generalize the method in modeling and controlling more complex systems. One critical challenge in applying the non-monotone pair-copula is the computational issue since the non-monotone pair-copula is typically piece-wise constructed with limited distribution families, where the numerical integration may be involved and the efficiency of the algorithm would be a potential concern. Another possible topic is the simultaneous control of multiple similar systems, e.g., production lines for different

types of food or pharmaceutical. These systems typically share most of the production features and perform in a similar pattern. The control over these similar systems can be greatly simplified if we can capture the similarity and identify the unique difference among the systems. This requires a multi-task modeling and control framework that extracts the production knowledge in each system, where our proposed method can serve as a sub-module for individual system modeling and control. We will study these topics in future work.

5.5 Appendix

5.5.1 Proof of lemma 3

We give a proof sketch of the equivalence between Equation 5.1 and Equation 5.8. Please note the $\boldsymbol{x}_I$ is conditioned and treated as a constant. As a result, it is general enough to consider a cost function with $\boldsymbol{x}_O$ only.

Based on Equation 5.5 and 5.7, we have:

$$\boldsymbol{x}_O|\boldsymbol{x}_I = T^{-1}\Big[h\Big(T(\boldsymbol{x}_I), \boldsymbol{\xi}\Big)\Big] \tag{5.16}$$

where T and T^{-1} are the general transformation and inverse transformation between $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$, h is the general form of Equation 5.7. Since T, T^{-1} and h are all strictly monotonic, we define a strictly monotonic function l with inverse function v as follows:

$$\begin{aligned} \boldsymbol{x}_O|\boldsymbol{x}_I &= l(\boldsymbol{x}_I, \boldsymbol{\xi}) \\ \boldsymbol{\xi}|\boldsymbol{x}_I &= v(\boldsymbol{x}_O, \boldsymbol{x}_I) \end{aligned} \tag{5.17}$$

As a result, we have

$$\mathrm{E}_{\boldsymbol{x}_O|\boldsymbol{x}_I} g(\boldsymbol{x}_O) = \int g(\boldsymbol{x}_O) f_{\boldsymbol{x}_O|\boldsymbol{x}_I}(\boldsymbol{x}_O) \mathrm{d}\boldsymbol{x}_O \tag{5.18}$$

where $f_{\boldsymbol{x}_O|\boldsymbol{x}_I}(\boldsymbol{x}_O)$ is the pdf of $\boldsymbol{x}_O|\boldsymbol{x}_I$. Please note that

$$f_{\boldsymbol{x}_O|\boldsymbol{x}_I}(\boldsymbol{x}_O) = f_{\boldsymbol{\xi}|\boldsymbol{x}_I}\Big(v(\boldsymbol{x}_O, \boldsymbol{x}_I)\Big) \cdot v'(\boldsymbol{x}_O, \boldsymbol{x}_I) \mathrm{d}\boldsymbol{x}_O \tag{5.19}$$

where $f_{\boldsymbol{\xi}|\boldsymbol{x}_I}$ is the pdf of $\boldsymbol{\xi}|\boldsymbol{x}_I$ and v' is the first order derivative towards $\boldsymbol{x}_O|\boldsymbol{x}_I$. Further notice that $\boldsymbol{\xi}$ and $\boldsymbol{x}_I$ are independent, then we have

$$\mathrm{E}_{\boldsymbol{x}_O|\boldsymbol{x}_I} g(\boldsymbol{x}_O) = \int g\bigg(T^{-1}\Big[h\big(T(\boldsymbol{x}_I), \boldsymbol{\xi}\big)\Big]\bigg) f_{\boldsymbol{\xi}}(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} = \mathrm{E}_{\boldsymbol{\xi}} g\bigg(T^{-1}\Big[h\big(T(\boldsymbol{x}_I), \boldsymbol{\xi}\big)\Big]\bigg) \tag{5.20}$$

Equation 5.20 demonstrates the Equation 5.1 and 5.8 are equivalent.

6 RESEARCH WORK TO DATE AND FUTURE WORK

The unprecedented data availability from smart and connected systems poses both opportunities and challenges for data analytics and decision making for operations management. This dissertation focuses on three specific challenges: i) Information sharing under heterogeneous systems. ii) Inter-relationship modeling and inference of non-linear association data under non-Gaussian distribution. iii) Efficient real time anomaly detection in smart and connected system. iv) Efficient control for minimizing the system operation cost under complex variable relationship.The contribution for each of the work is conclude as follows:

1 *Transfer Learning of Structures of Ordered Block Graphical Models Using Informative Priors*

In this study, we propose a structure learning method for ordered block graphical models. This is a score based method and mainly uses the Bayesian score framework that consists of the likelihood term and the structure prior term. The key innovation in this method is that we establish an informative structure prior by merging different historical graphs into the structure prior term under the Bayesian framework. The informative structure prior is expected to use historical information to benefit new graph structure learning. The key challenge of knowledge transferring in this work is the inconsistency of the sample space between historical processes and the new process.We propose a strategy by marginalizing inconsistent nodes and adding pseudo nodes to make the sample space consistent. A correcting algorithm based on the Bayes factor is also proposed to deal with the non-conformity between the prior knowledge and the new graph. We investigate the robustness of the proposed method

and find that the proposed method performs better than the method using non-informative priors in most practical cases and no worse than the non-informative prior based method in general.

2 *Approximate Key Performance Indicator Joint Distribution through Ordered Block Model and Pair Copula Construction*

We propose to approximate the KPIs joint distribution through sampling from the OBM-PCC model. First, the KPIs are modeled as random variables and the process is formulated under the OBM framework. Then, the pair-copulas are constructed based on the OBM structure to approximate the KPIs joint distribution. The feature of the proposed method is that it focuses on the distribution level approximation rather than the individual KPI estimation, which offers the feasibility in high dimensional modeling for KPIs under any configurations. Moreover, compared with traditional simulation method, the proposed method is more efficient due to the closed-form sampling procedures. The effectiveness and accuracy of the proposed method are verified through various numerical studies.

3 *Contamination Source Identification Based on Sequential Bayesian Approach for Water Distribution Network with Stochastic Demands*

In this work, we propose a sequential Bayesian scheme for contamination source identification in a water distribution network. This method treats a sensor alarm as available information and takes advantage of the sequence of alarms and time delay between alarms to perform real-time identification of a contamination source. The posterior probabilities of a node being the source of the contamination is calculated and used to make decisions. Moreover, a variation analysis of the obtained posterior probability is

conducted to obtain the significance probability of the obtained result. Our method considers hydraulic information and stochastic water demands in the water distribution network, which is less restrictive in practice. The effectiveness of the method is verified by a case study in EPANET with a 126-node water distribution network.

4 *Control of Key Performance Indicators of Manufacturing Production Systems through Pair Copula Modeling and Stochastic Optimization*

In this chapter, we propose to integrate the OBM-PCC model into the KPI control in general queuing systems. The KPI control is essentially an endogenous stochastic optimization problem, where the distribution of noise variable in the cost function depends on the decision variables. The OBM-PCC model is a flexible surrogate model for approximating the system input parameter and response KPI relationship in queuing systems. We investigate the properties of this relationship and develop a KPI control scheme that assimilates the unique feature of the OBM-PCC model, which degenerates the endogenous stochastic optimization into the traditional optimization problem. The key contribution of the chapter is that we employ the standard uniform distribution to link the OBM-PCC model and the KPI control problem so that the endogenous stochastic optimization in KPI control can be solved using traditional solvers. Moreover, the developed method is validated with extensive simulation studies to show the robustness towards the cost function, which is another benefit of the algorithm resulting from the integration of OBM-PCC model and the KPI control problem. The proposed method is compared with Kriging based methods in KPI control, and the results

demonstrate the effectiveness and superiority of our method.

Recent innovations across all disciplines have converged to make smart and connected systems technically/economically feasible. However, new methods are urgently needed for data analytics and decision making for operations management to complete this transformation towards smart and connected systems. I will continue to investigate novel data-driven based decision making methodologies and apply these techniques into emerging areas with smart and connected products/systems. More specifically, I will focus on:

1 *Information transfer for smart and connected system with big data*

Building upon my research accomplishments, I will continue to investigate novel data-driven methods to enable flexible and efficient transfer learning methodologies by integrating various stochastic processes and graphical models. In addition, sophisticated data-level fusion will be investigated to extract useful information from prodigious amount of data.

2 *Data-driven reinforcement learning for optimal management of smart and connected system*

The reinforcement learning can learn from the feedback signal and build a virtuous circle to balance the system performance and cost. The availability of data in smart and connected systems provide natural advantages for reinforcement learning, while the system uncertainty and complexity stump the learning. I will be dedicated on reinforcement learning with other data-driven techniques for better managing the smart and connected system.

The achievements of the proposed work will be reported in the future.

BIBLIOGRAPHY

[1] M. E. Porter and J. E. Heppelmann, "How smart, connected products are transforming competition," *Harvard business review*, vol. 92, no. 11, pp. 64–88, 2014.

[2] J. Bradley, J. Barbier, and D. Handler, "Embracing the internet of everything to capture your share of $14.4 trillion," *White Paper, Cisco*, 2013.

[3] P. Willett, P. F. Swaszek, and R. S. Blum, "The good, bad and ugly: distributed detection of a known signal in dependent gaussian noise," *IEEE Transactions on Signal Processing*, vol. 48, no. 12, pp. 3266–3279, 2000.

[4] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[5] V. Mansinghka, C. Kemp, T. Griffiths, and J. Tenenbaum, "Structured priors for structure learning," *arXiv preprint arXiv:1206.6852*, 2012.

[6] A. Bernard and A. J. Hartemink, "Informativestructure priors: Joint learning of dynamic," *Biocomputing 2005*, pp. 459–470, 2004.

[7] J. Z. Huang, N. Liu, M. Pourahmadi, and L. Liu, "Covariance matrix selection and estimation via penalised normal likelihood," *Biometrika*, pp. 85–98, 2006.

[8] L. Zeng and S. Zhou, "Inferring the interactions in complex manufacturing processes using graphical models," *Technometrics*, vol. 49, no. 4, pp. 373–381, 2007.

[9] J. Shi and S. Zhou, "Quality control and improvement for multistage systems: A survey," *IIE Transactions*, vol. 41, no. 9, pp. 744–753, 2009.

[10] L. M. d. Campos, "Independency relationships and learning algorithms for singly connected networks," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 10, no. 4, pp. 511–549, 1998.

[11] L. M. De Campos and J. F. Huete, "A new approach for learning belief networks using independence criteria," *International Journal of Approximate Reasoning*, vol. 24, no. 1, pp. 11–37, 2000.

[12] P. Spirtes, C. N. Glymour, and R. Scheines, *Causation, prediction, and search*. MIT press, 2000.

[13] F. Huang and S. Chen, "Learning dynamic conditional gaussian graphical models," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 4, pp. 703–716, 2018.

[14] G. Song, L. Han, and K. Xie, "Overlapping decomposition for gaussian graphical modeling," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 8, pp. 2217–2230, 2015.

[15] G. F. Cooper and E. Herskovits, "A bayesian method for the induction of probabilistic networks from data," *Machine learning*, vol. 9, no. 4, pp. 309–347, 1992.

[16] D. Heckerman, D. Geiger, and D. M. Chickering, "Learning bayesian networks: The combination of knowledge and statistical data," *Machine learning*, vol. 20, no. 3, pp. 197–243, 1995.

[17] J. Peters and P. Bühlmann, "Identifiability of gaussian structural equation models with equal error variances," *Biometrika*, vol. 101, no. 1, pp. 219–228, 2013.

[18] M. Teyssier and D. Koller, "Ordering-based search: A simple and effective algorithm for learning bayesian networks," *arXiv preprint arXiv:1207.1429*, 2012.

[19] W. B. Wu and M. Pourahmadi, "Nonparametric estimation of large covariance matrices of longitudinal data," *Biometrika*, pp. 831–844, 2003.

[20] P. Holmström, *Modelling manufacturing systems capability.* PhD thesis, KTH, 2006.

[21] T. J. Koski, J. M. Noble, and F. L. Rios, "The minimal hoppe-beta prior distribution for directed acyclic graphs and structure learning," 2016.

[22] M. J. Flores, A. E. Nicholson, A. Brunskill, K. B. Korb, and S. Mascaro, "Incorporating expert knowledge when learning bayesian network structure: a medical case study," *Artificial intelligence in medicine*, vol. 53, no. 3, pp. 181–204, 2011.

[23] D. Geiger and D. Heckerman, "Learning gaussian networks," in *Proceedings of the Tenth international conference on Uncertainty in artificial intelligence*, pp. 235–243, Morgan Kaufmann Publishers Inc., 1994.

[24] N. Friedman, I. Nachman, and D. Peér, "Learning bayesian network structure from massive datasets: the 'sparse candidate' algorithm," in *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pp. 206–215, Morgan Kaufmann Publishers Inc., 1999.

[25] B. Jian and B. C. Vemuri, "Robust point set registration using gaussian mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1633–1645, 2011.

[26] J.-M. Marin, K. Mengersen, and C. P. Robert, "Bayesian modelling and inference on mixtures of distributions," *Handbook of statistics*, vol. 25, pp. 459–507, 2005.

[27] S. Yang and K.-C. Chang, "Comparison of score metrics for bayesian network learning," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 32, no. 3, pp. 419–428, 2002.

[28] Z. Liu, B. Malone, and C. Yuan, "Empirical evaluation of scoring functions for bayesian network model selection," *BMC bioinformatics*, vol. 13, no. 15, p. S14, 2012.

[29] J. Whittaker, *Graphical models in applied multivariate statistics*. Wiley Publishing, 2009.

[30] H. J. Hung, R. T. O'Neill, P. Bauer, and K. Kohne, "The behavior of the p-value when the alternative hypothesis is true," *Biometrics*, pp. 11–22, 1997.

[31] A. F. Smith, "A general bayesian linear model," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 67–75, 1973.

[32] J. O. Berger and L. R. Pericchi, "The intrinsic bayes factor for model selection and prediction," *Journal of the American Statistical Association*, vol. 91, no. 433, pp. 109–122, 1996.

[33] S. Huang, J. Li, J. Ye, A. Fleisher, K. Chen, T. Wu, E. Reiman, A. D. N. Initiative, *et al.*, "A sparse structure learning algorithm for gaussian bayesian network identification from high-dimensional data," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 6, pp. 1328–1342, 2013.

[34] J.-P. Loose, N. Chen, and S. Zhou, "Surrogate modeling of dimensional variation propagation in multistage assembly processes," *IIE Transactions*, vol. 41, no. 10, pp. 893–904, 2009.

[35] J.-P. Loose, Q. Zhou, S. Zhou, and D. Ceglarek, "Integrating gd&t into dimensional variation models for multistage machining processes," *International Journal of Production Research*, vol. 48, no. 11, pp. 3129–3149, 2010.

[36] N. Friedman, M. Goldszmidt, and A. Wyner, "Data analysis with bayesian networks: A bootstrap approach," in *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pp. 196–205, Morgan Kaufmann Publishers Inc., 1999.

[37] K. Zhang, J. Peters, D. Janzing, and B. Schölkopf, "Kernel-based conditional independence test and application in causal discovery," *arXiv preprint arXiv:1202.3775*, 2012.

[38] N. Kang, C. Zhao, J. Li, and J. A. Horst, "A hierarchical structure of key performance indicators for operation management and continuous improvement in production systems," *International Journal of Production Research*, vol. 54, no. 21, pp. 6333–6350, 2016.

[39] R. Kontar, S. Zhou, and J. Horst, "Estimation and monitoring of key performance indicators of manufacturing systems using the multi-output gaussian process," *International Journal of Production Research*, vol. 55, no. 8, pp. 2304–2319, 2017.

[40] R. R. Rodriguez, J. J. A. Saiz, and A. O. Bas, "Quantitative relationships between key performance indicators for supporting decision-making processes," *Computers in Industry*, vol. 60, no. 2, pp. 104–113, 2009.

[41] J. A. Buzacott and J. G. Shanthikumar, *Stochastic models of manufacturing systems*, vol. 4. Prentice Hall Englewood Cliffs, NJ, 1993.

[42] G. J. Hahn, T. Sens, C. Decouttere, and N. J. Vandaele, "A multi-criteria approach to robust outsourcing decision-making in stochastic manufacturing systems," *Computers & Industrial Engineering*, vol. 98, pp. 275–288, 2016.

[43] J. G. Shanthikumar, S. Ding, and M. T. Zhang, "Queueing theory for semiconductor manufacturing systems: a survey and open problems," *IEEE Transactions on Automation Science and Engineering*, vol. 4, no. 4, pp. 513–522, 2007.

[44] K. Wu and L. McGinnis, "Interpolation approximations for queues in series," *IIE transactions*, vol. 45, no. 3, pp. 273–290, 2013.

[45] M.-G. Huang, P.-L. Chang, and Y.-C. Chou, "Analytic approximations for multiserver batch-service workstations with multiple process recipes in semiconductor wafer fabrication," *IEEE Transactions on Semiconductor Manufacturing*, vol. 14, no. 4, pp. 395–405, 2001.

[46] S. Balsamo, V. de Nitto Personé, and R. Onvural, *Analysis of queueing networks with blocking*, vol. 31. Springer Science & Business Media, 2013.

[47] N. Chen and S. Zhou, "Simulation-based estimation of cycle time using quantile regression," *IIE Transactions*, vol. 43, no. 3, pp. 176–191, 2010.

[48] D. Cooley, R. A. Davis, and P. Naveau, "The pairwise beta distribution: A flexible parametric multivariate model for extremes," *Journal of Multivariate Analysis*, vol. 101, no. 9, pp. 2103–2117, 2010.

[49] S. Cavanillas, J. M. Díaz-Cruz, C. Ariño, and M. Esteban, "Parametric signal fitting by gaussian peak adjustment: A new multivariate curve resolution method for non-bilinear voltammetric measurements," *Analytica chimica acta*, vol. 689, no. 2, pp. 198–205, 2011.

[50] D. W. Scott, *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

[51] B. McCune, "Non-parametric habitat models with automatic interactions," *Journal of Vegetation Science*, vol. 17, no. 6, pp. 819–830, 2006.

[52] T.-S. Lee and I.-F. Chen, "A two-stage hybrid credit scoring model using artificial neural networks and multivariate adaptive regression splines," *Expert Systems with Applications*, vol. 28, no. 4, pp. 743–752, 2005.

[53] T. Chen, J. Morris, and E. Martin, "Gaussian process regression for multivariate spectroscopic calibration," *Chemometrics and Intelligent Laboratory Systems*, vol. 87, no. 1, pp. 59–71, 2007.

[54] R. B. Nelsen, "Introduction," in *An Introduction to Copulas*, pp. 1–4, Springer, 1999.

[55] H. Joe, "Families of m-variate distributions with given margins and m (m-1)/2 bivariate dependence parameters," *Lecture Notes-Monograph Series*, pp. 120–141, 1996.

[56] C. Czado, "Pair-copula constructions of multivariate copulas," *Copula theory and its applications*, pp. 93–109, 2010.

[57] A. Bauer, C. Czado, and T. Klein, "Pair-copula constructions for non-gaussian dag models," *Canadian Journal of Statistics*, vol. 40, no. 1, pp. 86–109, 2012.

[58] P. Billingsley, *Probability and measure*. John Wiley & Sons, 2008.

[59] P. Heidelberger and P. A. Lewis, "Quantile estimation in dependent sequences," *Operations Research*, vol. 32, no. 1, pp. 185–209, 1984.

[60] F. Yang, B. E. Ankenman, and B. L. Nelson, "Estimating cycle time percentile curves for manufacturing systems via simulation," *INFORMS Journal on Computing*, vol. 20, no. 4, pp. 628–643, 2008.

[61] A. Sklar, "Fonctions de répartition à n dimensions et leurs marges. publications de l'institut de statistique de l'université de paris," 1959.

[62] T. Bedford and R. M. Cooke, "Probability density decomposition for conditionally dependent random variables modeled by vines," *Annals of Mathematics and Artificial intelligence*, vol. 32, no. 1, pp. 245–268, 2001.

[63] T. Bedford and R. M. Cooke, "Vines: A new graphical model for dependent random variables," *Annals of Statistics*, pp. 1031–1068, 2002.

[64] K. Aas, C. Czado, A. Frigessi, and H. Bakken, "Pair-copula constructions of multiple dependence," *Insurance: Mathematics and economics*, vol. 44, no. 2, pp. 182–198, 2009.

[65] U. Schepsmeier, "Maximum likelihood estimation of c-vine pair-copula constructions based on bivariate copulas from different families," 2010.

[66] A. Min and C. Czado, "Bayesian model selection for d-vine pair-copula constructions," *Canadian Journal of Statistics*, vol. 39, no. 2, pp. 239–258, 2011.

[67] E. Pircalabelu, G. Claeskens, and I. Gijbels, "Copula directed acyclic graphs," *Statistics and Computing*, vol. 27, no. 1, pp. 55–78, 2017.

[68] A. Hanea, O. M. Napoles, and D. Ababei, "Non-parametric bayesian networks: Improving theory and reviewing applications," *Reliability Engineering & System Safety*, vol. 144, pp. 265–284, 2015.

[69] H. Akaike, "A new look at the statistical model identification," *IEEE transactions on automatic control*, vol. 19, no. 6, pp. 716–723, 1974.

[70] E. Brechmann, "Truncated and simplified regular vines and their applications," 2010.

[71] D. Schirmacher and E. Schirmacher, "Multivariate dependence modeling using pair-copulas," tech. rep., Technical report, 2008.

[72] F. J. Massey Jr, "The kolmogorov-smirnov test for goodness of fit," *Journal of the American statistical Association*, vol. 46, no. 253, pp. 68–78, 1951.

[73] J. H. Drew, A. G. Glen, and L. M. Leemis, "Computing the cumulative distribution function of the kolmogorov–smirnov statistic," *Computational statistics & data analysis*, vol. 34, no. 1, pp. 1–15, 2000.

[74] J. Geweke *et al.*, *Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments*, vol. 196. Federal Reserve Bank of Minneapolis, Research Department Minneapolis, MN, USA, 1991.

[75] A. Gelman and D. B. Rubin, "Inference from iterative simulation using multiple sequences," *Statistical science*, pp. 457–472, 1992.

[76] T. Mikosch, "How to model multivariate extremes if one must?," *Statistica Neerlandica*, vol. 59, no. 3, pp. 324–338, 2005.

[77] B. Renard and M. Lang, "Use of a gaussian copula for multivariate extreme value analysis: some case studies in hydrology," *Advances in Water Resources*, vol. 30, no. 4, pp. 897–912, 2007.

[78] A. Dvoretzky, J. Kiefer, and J. Wolfowitz, "Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator," *The Annals of Mathematical Statistics*, pp. 642–669, 1956.

[79] A. Lee, "Generating random binary deviates having fixed marginal distributions and specified degrees of association," *The American Statistician*, vol. 47, no. 3, pp. 209–215, 1993.

[80] K. A. Nilsson, S. G. Buchberger, and R. M. Clark, "Simulating exposures to deliberate intrusions into water distribution systems," *Journal of water resources planning and management*, vol. 131, no. 3, pp. 228–236, 2005.

[81] J. Qiao, D. Jeong, M. Lawley, J.-P. P. Richard, D. M. Abraham, and Y. Yih, "Allocating security resources to a water supply network," *IIE Transactions*, vol. 39, no. 1, pp. 95–109, 2007.

[82] A. Ostfeld and E. Salomons, "Optimal early warning monitoring system layout for water networks security: inclusion of sensors sensitivities and response delays," *Civil Engineering and Environmental Systems*, vol. 22, no. 3, pp. 151–169, 2005.

[83] E. Todini and S. Pilati, "A gradient algorithm for the analysis of pipe networks," in *Computer applications in water supply: vol. 1—systems analysis and simulation*, pp. 1–20, Research Studies Press Ltd., 1988.

[84] A. E. De Sanctis, F. Shang, and J. G. Uber, "Real-time identification of possible contamination sources using network backtracking methods," *Journal of Water Resources Planning and Management*, vol. 136, no. 4, pp. 444–453, 2009.

[85] C. D. Laird, L. T. Biegler, and B. G. van Bloemen Waanders, "Mixed-integer approach for obtaining unique solutions in source inversion of water networks," *Journal of Water Resources Planning and Management*, vol. 132, no. 4, pp. 242–251, 2006.

[86] C. D. Laird, L. T. Biegler, B. G. van Bloemen Waanders, and R. A. Bartlett, "Contamination source determination for water networks," *Journal of Water Resources Planning and Management*, vol. 131, no. 2, pp. 125–134, 2005.

[87] M. S. Dehghani and H. D. Sherali, "A resource allocation approach for managing critical network-based infrastructure systems," *IIE Transactions*, vol. 48, no. 9, pp. 826–837, 2016.

[88] K. Lansey, D. Delgado, and Q. Banihani, "Discussion of "mixed-integer approach for obtaining unique solutions in source inversion of water networks" by carl d. laird, lorenz t. biegler, and bart g. van bloemen waanders," *Journal of Water Resources Planning and Management*, vol. 133, no. 6, pp. 573–575, 2007.

[89] F. Shang, J. G. Uber, and M. M. Polycarpou, "Particle backtracking algorithm for water distribution system analysis," *Journal of environmental engineering*, vol. 128, no. 5, pp. 441–450, 2002.

[90] A. De Sanctis, D. Boccelli, F. Shang, and J. Uber, "Probabilistic approach to characterize contamination sources with imperfect sensors," in *World Environmental and Water Resources Congress 2008: Ahupua'A*, pp. 1–10, 2008.

[91] Z.-x. Liu and W.-x. Xie, "Multi-target bayesian filter for propagating marginal distribution," *Signal Processing*, vol. 105, pp. 328–337, 2014.

[92] Z.-x. Liu, L.-j. Li, W.-x. Xie, and L.-q. Li, "Two implementations of marginal distribution bayes filter for nonlinear gaussian models," *AEU-International Journal of Electronics and Communications*, vol. 69, no. 9, pp. 1297–1304, 2015.

[93] Z.-x. Liu, L.-j. Li, W.-x. Xie, and L.-q. Li, "Sequential measurement-driven multi-target bayesian filter," *EURASIP Journal on Advances in Signal Processing*, vol. 2015, no. 1, p. 43, 2015.

[94] C. D. Cristo and A. Leopardi, "Pollution source identification of accidental contamination in water distribution networks," *Journal of Water Resources Planning and Management*, vol. 134, no. 2, pp. 197–202, 2008.

[95] A. Preis and A. Ostfeld, "Contamination source identification in water systems: A hybrid model trees–linear programming scheme," *Journal of Water Resources Planning and Management*, vol. 132, no. 4, pp. 263–273, 2006.

[96] F. Shang, J. Uber, and R. Murray, "Uncertainty of contamination source characterization within water distribution systems," in *World Environmental and Water Resources Congress 2007: Restoring Our Natural Habitat*, pp. 1–8, 2007.

[97] P. Van Thienen, D. Vries, B. De Graaf, M. Van de Roer, P. Schaap, and E. Zaadstra, "Probabilistic backtracing of drinking water contamination events in a stochastic world," *Procedia Engineering*, vol. 70, pp. 1688–1696, 2014.

[98] X. Yang and D. L. Boccelli, "Bayesian approach for real-time probabilistic contamination source identification," *Journal of Water Resources Planning and Management*, vol. 140, no. 8, p. 04014019, 2013.

[99] L. Bian and N. Gebraeel, "Stochastic modeling and real-time prognostics for multi-component systems with degradation rate interactions," *IIE Transactions*, vol. 46, no. 5, pp. 470–482, 2014.

[100] N. Z. Gebraeel, M. A. Lawley, R. Li, and J. K. Ryan, "Residual-life distributions from component degradation signals: A bayesian approach," *IiE Transactions*, vol. 37, no. 6, pp. 543–557, 2005.

[101] W. J. Dawsey, B. S. Minsker, and V. L. VanBlaricum, "Bayesian belief networks to integrate monitoring evidence of water distribution system contamination," *Journal of Water Resources Planning and Management*, vol. 132, no. 4, pp. 234–241, 2006.

[102] H. Wang and K. W. Harrison, "Bayesian update method for contaminant source characterization in water distribution systems," *Journal of Water Resources Planning and Management*, vol. 139, no. 1, pp. 13–22, 2011.

[103] H. Wang and K. W. Harrison, "Improving efficiency of the bayesian approach to water distribution contaminant source characterization with support vector regression," *Journal of Water Resources Planning and Management*, vol. 140, no. 1, pp. 3–11, 2012.

[104] G. Wang, S. H. Huang, and J. P. Dismukes, "Product-driven supply chain selection using integrated multi-criteria decision-making methodology," *International journal of production economics*, vol. 91, no. 1, pp. 1–15, 2004.

[105] J. Luo and M. A. Nascimento, "Content based sub-image retrieval via hierarchical tree matching," in *Proceedings of the 1st ACM international workshop on Multimedia databases*, pp. 63–69, ACM, 2003.

[106] H. Kang, S. J. Yoo, and D. Han, "Modeling web crawler wrappers to collect user reviews on shopping mall with various hierarchical tree structure," in *Web Information Systems and Mining, 2009. WISM 2009. International Conference on*, pp. 69–73, IEEE, 2009.

[107] J. M. Hellawell, *Biological indicators of freshwater pollution and environmental management*. Springer Science & Business Media, 2012.

[108] I. Basupi and Z. Kapelan, "Evaluating flexibility in water distribution system design under future demand uncertainty," *Journal of Infrastructure Systems*, vol. 21, no. 2, p. 04014034, 2014.

[109] L. Liu, A. Sankarasubramanian, and S. R. Ranjithan, "Logistic regression analysis to estimate contaminant sources in water distribution systems," *Journal of Hydroinformatics*, vol. 13, no. 3, pp. 545–557, 2011.

[110] A. K. Gupta and S. Nadarajah, *Handbook of beta distribution and its applications*. CRC press, 2004.

[111] L. A. Rossman *et al.*, "Epanet 2: users manual," 2000.

[112] A. Ostfeld, J. G. Uber, E. Salomons, J. W. Berry, W. E. Hart, C. A. Phillips, J.-P. Watson, G. Dorini, P. Jonkergouw, Z. Kapelan, *et al.*, "The battle of the water sensor networks (bwsn): A design challenge for engineers and algorithms," *Journal of Water Resources Planning and Management*, vol. 134, no. 6, pp. 556–568, 2008.

[113] "Centre of water systems." `http://socialsciences.exeter.ac.uk/politics/research/centres/ceg`, 2007.

[114] C. Z. Mooney, R. D. Duval, and R. Duvall, *Bootstrapping: A nonparametric approach to statistical inference*. No. 94-95, Sage, 1993.

[115] S. Stidham Jr and R. R. Weber, "Monotonic and insensitive optimal policies for control of queues with undiscounted costs," *Operations research*, vol. 37, no. 4, pp. 611–625, 1989.

[116] A. Bassamboo, J. M. Harrison, and A. Zeevi, "Design and control of a large call center: Asymptotic analysis of an lp-based method," *Operations Research*, vol. 54, no. 3, pp. 419–435, 2006.

[117] Z. Aksin, M. Armony, and V. Mehrotra, "The modern call center: A multi-disciplinary perspective on operations management research," *Production and operations management*, vol. 16, no. 6, pp. 665–688, 2007.

[118] G. Giambene, *Queuing theory and telecommunications*. Springer, 2005.

[119] H. Shen, L. J. Hong, and X. Zhang, "Enhancing stochastic kriging for queueing simulation with stylized models," *IISE Transactions*, vol. 50, no. 11, pp. 943–958, 2018.

[120] J. M. Bekki, X. Chen, and D. Batur, "Steady-state quantile parameter estimation: an empirical comparison of stochastic kriging and quantile regression," in *Proceedings of the 2014 winter simulation conference*, pp. 3880–3891, IEEE Press, 2014.

[121] X. Chen, B. E. Ankenman, and B. L. Nelson, "Enhancing stochastic kriging metamodels with gradient estimators," *Operations Research*, vol. 61, no. 2, pp. 512–528, 2013.

[122] B. Ankenman, B. L. Nelson, and J. Staum, "Stochastic kriging for simulation metamodeling," *Operations research*, vol. 58, no. 2, pp. 371–382, 2010.

[123] C. Wang and S. Zhou, "Approximate multivariate distribution of key performance indicators through ordered block model and pair-copula construction," *IISE Transactions*, no. just-accepted, pp. 1–30, 2018.

[124] L. A. Hannah, "Stochastic optimization," *International Encyclopedia of the Social & Behavioral Sciences*, vol. 2, pp. 473–481, 2015.

[125] H. Jalali, I. Van Nieuwenhuyse, and V. Picheny, "Comparison of kriging-based algorithms for simulation optimization with heterogeneous noise," *European Journal of Operational Research*, vol. 261, no. 1, pp. 279–301, 2017.

[126] M. Chau, M. C. Fu, H. Qu, and I. O. Ryzhov, "Simulation optimization: a tutorial overview and recent developments in gradient-based methods," in *Proceedings of the Winter Simulation Conference 2014*, pp. 21–35, IEEE, 2014.

[127] T. Kirschenmann, E. Popova, P. Damien, and T. Hanson, "Decision dependent stochastic processes," *European Journal of Operational Research*, vol. 234, no. 3, pp. 731–742, 2014.

[128] B. Tarhan, I. E. Grossmann, and V. Goel, "Stochastic programming approach for the planning of offshore oil or gas field infrastructure under decision-dependent uncertainty," *Industrial & Engineering Chemistry Research*, vol. 48, no. 6, pp. 3078–3097, 2009.

[129] Y. Zhan, Q. P. Zheng, J. Wang, and P. Pinson, "Generation expansion planning with large amounts of wind power via decision-dependent stochastic programming," *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 3015–3026, 2016.

[130] J. C. Spall, "Implementation of the simultaneous perturbation algorithm for stochastic optimization," *IEEE Transactions on aerospace and electronic systems*, vol. 34, no. 3, pp. 817–823, 1998.

[131] T. Mikosch, "Copulas: Tales and facts," *Extremes*, vol. 9, no. 1, pp. 3–20, 2006.

[132] J. C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE transactions on automatic control*, vol. 37, no. 3, pp. 332–341, 1992.

[133] D. Huang, T. T. Allen, W. I. Notz, and N. Zeng, "Global optimization of stochastic black-box systems via sequential kriging meta-models," *Journal of global optimization*, vol. 34, no. 3, pp. 441–466, 2006.

[134] J. A. Nelder and R. Mead, "A simplex method for function minimization," *The computer journal*, vol. 7, no. 4, pp. 308–313, 1965.