2015

A Metamodel-Based Monte Carlo Simulation Approach for Responsive Production Planning of Manufacturing Systems

Minqi Li

A Metamodel-Based Monte Carlo Simulation Approach for Responsive Production Planning of Manufacturing Systems

Minqi Li

Dissertation submitted to the
Statler College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy
in
Industrial Engineering

Feng Yang, Ph.D., Chair
Wafik H. Iskander, Ph.D.
Majid Jaridi, Ph.D.
Bhaskaran Gopalakrishnan, Ph.D.
Robert Mnatsakanov, Ph.D.

Department of Industrial and Management Systems Engineering

Morgantown, West Virginia
2015

Keywords: Metamodel, Non-Stationary Autoregressive Model, Bivariate Time-Series Model, Monte Carlo Simulation, Production Planning

ABSTRACT

A Metamodel-Based Monte Carlo Simulation Approach for
Responsive Production Planning of Manufacturing Systems

Minqi Li

Production planning is concerned with finding a release plan of jobs into the manufacturing system so that its actual outputs over time match the customer demand with the least cost. The biggest challenge of production planning lies in the difficulty to quantify the performance of a release plan, which is the necessary basis for plan optimization. Triggered by an input plan over a time horizon, the system outputs, work in process (WIP) and job departures, are non-stationary bivariate time series that interact with customer demand (another time series), resulting in the fulfillment/non-fulfillment of demand and in the holding cost of both WIP and finished-goods inventory. The relationship between a release plan and its resulting performance metrics (typically, mean/variance of the total cost and the demand fulfill rate) is far from being adequately quantified in the existing literature of production planning. In this dissertation, a metamodel-based Monte Carlo simulation (MCS) method is developed to accurately capture the dynamic and stochastic behavior of a manufacturing system, and to allow for real-time evaluation of a release plan in terms of its performance metrics. This evaluation capability is embedded in a multi-objective optimization framework to enable the quick search of good (or optimum) release plans. The developed method has been applied to a scaled-down semiconductor fabrication system to demonstrate the quality of the metamodel-based MCS evaluation and the plan optimization results.

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Feng Yang, for giving me the opportunity to work on this project, and for her support throughout my study. It is my great honor to work under her guidance. This dissertation could not have been written without her talent, encouragement, patient and financial support. I am also thankful to Dr. Iskander, Dr. Jaridi, Dr. Gopalakrishnan, and Dr. Mnatsakanov for being my committee members and for their assistance in preparing this dissertation.

Contents

List of Figures

List of Tables

List of Symbols

$A(t)$	arrival process that counts the number of arrivals during $(t, t+1]$
$B(t, \mathbf{x})$	the amount of customer demand that cannot be satisfied at the end of the t^{th} time unit
$D(t)$	departure process that counts the number of finished jobs during $(t, t+1]$
$\mathcal{D}(t)$	demand process which counts the quantity requested by customers at the end of the t^{th} time unit
$d_i(t)$	the i^{th} moment of $D(t)$
$d_{1j}(t)$	the first moment of $D(t)D(t-j)$
$e_{QD}(t)$	the first moment of $Q(t)D(t)$
G_0	initial number of finished jobs or backlogged orders at time 0
$G(t, \mathbf{x})$	the number of finished jobs or backlogged orders at the end of the t^{th} time unit
H	length of planning horizon measured in basic time unit
$I(t, \mathbf{x})$	the number of inventory of finished jobs at the end of the t^{th} time unit
h_t	unit holding cost of finished jobs per time unit at time t
J_Q	highest order of the time lag needed to describe the WIP process
J_D	highest order of the time lag needed to describe the departure process
K	number of distinct levels used in the arrival rate function for the estimation data set
L	length of estimation data set for metamodel
$m_i(t)$	the i^{th} moment of $Q(t)$
$m_{1j}(t)$	the first moment of $Q(t)Q(t-j)$

N	number of observations collected
P	number of time periods in the planning horizon
p	index of time period in the planning horizon
$Q(t)$	WIP process that counts the number of work in process in the system at time t
R	number of replications in simulation
$DF(\mathbf{x})$	demand fulfill rate associated with a release plan $\mathbf{x}$
$TC(\mathbf{x})$	total cost associated with a release plan $\mathbf{x}$
t	time index measured in terms of the basic time unit
U	number of demand realizations over the planning horizon
w_t	unit holding cost of WIP per time unit at time t
$\mathbf{x}$	release plan of jobs
$x(t)$	arrival rate at time t
$\mathbf{y}(t)$	characteristics vector of system outputs at time t
$Z(t)$	a random process that represents either the WIP process or the departure process

List of Abbreviations

AR	Autoregressive
BOM	Bill of materials
CDF	Cumulative distribution functions
CF	Clearing function
DES	Discrete-event simulation
DOE	Design of experiments
EDS	Estimation data set
FG	Finished goods
GP	Generalized Poisson
INAR	Integer valued autoregressive
LP	Linear programming
MCS	Monte Carlo simulation
MOGA	Multi-objective genetic algorithm
MRP	Material requirement planning
N-AR	Non-stationary autoregressive model
N-INAR	Non-stationary integer valued autoregressive model
NSGA-II	Elitism Non-Dominated Sorting GA
N-SINAR	Non-stationary signed integer valued autoregressive model
OvS	Optimization via simulation
PT	Processing times
SPACF	Sample partial autocorrelation function
TTF	Times to failure

TTR	Times to repair
VDS	Validation data sets
WIP	Work in process

Chapter 1

Introduction

This dissertation is concerned with production planning in manufacturing, which can be loosely defined as the problem of finding a release schedule of jobs into the manufacturing system so that the actual outputs over time satisfy, as closely as possible, the predetermined requirements [71]. The planning horizon of production activities usually ranges from one or several months to two years, and the frequency of planning/replanning is weekly or monthly [40].

The purpose of production planning is to find the optimal release schedule of jobs so that the system's overall performance can be optimized. Typically, the planning horizon is divided into a number of time buckets (periods), and the decision variables are the quantities of jobs released into the system for processing during each time bucket. The performance metrics to be optimized usually include (i) the total cost (or sometimes profit), which may consist of the holding cost for finished goods (FG) and work in process (WIP) inventories, and (ii) the demand fulfill rate, which may be defined as the percentage of immediately satisfied demand.

Optimizing the performance metrics with respect to (w.r.t.) the release plan is challenging, simply because it is notoriously difficult to quantify the relationships between the performance metrics and the input decisions. A real manufacturing system is subject to inherent uncertainty such as probabilistic processing times, machine failures, etc. The existence of such uncertainty leads to the complicated input-output relationships of the system. Precise definitions will be given later in Chapter 3, and herein, the symbols $A(t)$, $Q(t)$, and

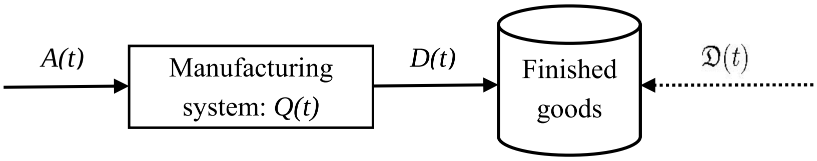


Figure 1.1: The input-output process of a manufacturing system.

$D(t)$ are loosely used to represent the three time series respectively: $A(t)$ denotes the number of jobs released for processing, $Q(t)$ the number of jobs (i.e., WIP) in the manufacturing system, and $D(t)$ the number of departure of completed jobs from the system. Figure 1.1 illustrates the input-output process of a manufacturing system. The release process $A(t)$ is determined by the decision variables (i.e., the release plan). Triggered by the input drive $A(t)$, which may well vary over time, $Q(t)$ and $D(t)$ are the non-stationary time series for the system's outputs, and their evolution also depends on the initial status of the system. The ultimate performance metrics depend on $Q(t)$, $D(t)$, and the customer demand $\mathcal{D}(t)$: The WIP holding cost is determined by $Q(t)$; the FG holding cost and the demand fulfill rate depend on the interaction between the departure process $D(t)$ and the customer demand $\mathcal{D}(t)$. In industrial practice, demand is generally a non-stationary time series as well, and is specified through the forecasting efforts exogenous to production planning.

Despite continuous research efforts, it remains a challenge to adequately quantify the dependence of the performance metrics upon the input release for responsive production planning, due to the triggering/interaction between the non-stationary time series: $A(t)$, $Q(t)$, $D(t)$, and $\mathcal{D}(t)$. To address this difficulty, a metamodel-based Monte Carlo simulation (MCS) method is developed in this dissertation which has the following features. First, for a given release plan, it enables the thorough evaluation of the probabilistic measures of the system performances, which include not only the expectations (e.g., the mean cost) but also the variances (e.g., the variance of the cost) and probabilities of interest (e.g., the demand fulfill rate). Second, it is able to accommodate practically any demand patterns. Third, it allows for a quick evaluation of a candidate plan in terms of its performance metrics, and provides the necessary basis for plan optimization in a timely manner.

The remainder of this dissertation is organized as follows. Chapter 2 provides a review of the existing literature. Chapter 3 gives an overview of the metamodel-based MCS method for responsive production planning. Chapter 4 details the input-output metamodeling of a manufacturing system, and the metamodel-based MCS is discussed in Chapter 5. Chapter 6 formulates the multi-objective optimization problem for production planning, and presents the optimization scheme with the MCS method embedded to quickly evaluate each candidate plan. In Chapter 7, the plan optimization approach is applied on a scaled-down semiconductor fabrication system for demonstration. A brief summary is given in Chapter 8.

Chapter 2

Literature Review

The existing methods of production planning can be divided into two categories, the mathematical programming, and optimization via simulation (OvS) methods, which will be discussed respectively in this chapter.

2.1 Mathematical Programming Methods

There is an extensive literature on mathematical programming models for production planning. We classify these works into three groups, depending on how the manufacturing uncertainty is addressed in the models.

The first group of methods completely disregard the uncertainty involved in manufacturing processes, and consider system outputs (WIP or flow time) as exogenous parameters independent of job releases into the system. The vast majority of linear and integer programming models (e.g., [48, 12, 38, 75]) fall into this group. These models are generally computationally tractable, but their accuracy is very much questionable, especially when the manufacturing system is heavily utilized. In this stream of works, some stochastic programming attempts have been made to accommodate demand uncertainty [32, 91, 41].

Recognizing the queueing effects caused by manufacturing uncertainty, substantial research efforts have been made to incorporate into mathematical programming models the relationships between system outputs and input releases. However, this second group of research relies on the assumption that the system is operated in steady state, to more or less extent. The iterative approach developed in [44, 18, 19, 43, 60] integrates a detailed discrete-event simulation (DES) model with a linear programming (LP) model to account for the dependence of system outputs upon job releases. The iterative approach starts by setting initial values of the job lead times and feeding them into the LP model to obtain a production plan. Then the production plan is realized by the DES model and a set of estimated job flow times are obtained which will serve as the values of job lead times used in the next iteration. The iterative scheme stops when the changes of the estimated flow times between iterations are relatively small such that certain convergence rule is satisfied. The empirical results of Hung and Leachman [44] has shown that the iterative scheme converges rapidly in most situations but may fail to converge in some cases which required further investigation. Irdem et al. [45] also indicates that this approach has a convergency problem when the system is working under heavy workload.

Into the second group also falls the clearing function(CF)-based methods [7, 8, 1, 45, 46, 52, 53, 54, 2], which have drawn a lot of attention. Clearing functions are regression models estimated from DES data seeking to capture a system's capacity of resource, and are included as constraints in the optimization formulation of production planning. The common approach to estimated the CF is to derive an analytical forms based on the steady-state queuing theory first and then estimate the unknown parameters involved via regression [57, 8]. Asmundsson et al. [7] also suggests visually fit the piecewise linear CF based on the data collected from simulation experiments. Built on the CF formulation, some recent efforts have been made to take into account random demand patterns by developing stochastic programming or chance constrained optimization methods [73, 5, 77, 6]. In the second group of work, although non-stationary DES data are frequently collected, they only affect the optimization results through the models (e.g., CF) or parameters fitted from them. Such model fitting implicitly assumes stationarity of the DES data, and hence the fitted models, which serve as given constraints for plan optimization, can at best provide a snapshot for the system's non-stationary behavior. The limitation of stationary approximation for generally non-stationary manufacturing systems has been long recognized and discussed [72, 81, 69, 70].

The third group includes a few research works to address the dynamic behavior of system outputs in mathematical programming for production planning [78, 69, 70, 39]. Based on transient Little's Law [11], Riano [78] developed an approximate algorithm to establish the transient relationships between expected WIP, departures, and job releases. This analytical approximation, which is closely tied to production planning, can be considered as parallel to the various approximation methods (e.g., fluid and diffusion approximations) for the transient analysis of non-Markovian queueing systems in the general queueing literature [22, 66, 59]; The analytical approximation methods are restricted to certain unrealistic assumptions, and are inadequate to fully accommodate many features of real manufacturing systems such as non-Markovian interarrival/service times, server failures, re-entrant job flows, etc. Missbauer [69] proposed a transient CF, which takes an exponential functional form; compared to its stationary counterparts, the transient CF includes the time factor as an additional predictor. Missbauer [70] developed a two-dimensional CF by assuming that for each workstation, the initial WIP at the beginning of a period and the total input of products during that period are random variables following certain joint probability distribution. However, as indicated by the author, these two works left a lot of questions and needed to be tested using empirical or simulation data. Haeussler and Missbauer [39] tested the performance of CF-based approaches with additional independent predictors (e.g., the expectation and variability of the WIP from the previous time periods) using a flexible flow shop and a scaled-down real manufacturing system.

As a final note, all the works reviewed in this section adopt a mathematical programming framework. In the formulation of the optimization problems, the WIP and job departures at each workstation are treated as deterministic variables, even though they may be considered as related to each other through some functional (stationary or non-stationary) relationships. Hence, the interaction between workstations (caused by the fact that the stochastic departures from an upstream station serve as the random arrivals to its downstream station) cannot be fully captured; similarly, the interaction between departures of completed jobs and customer demand, both of which are time series, cannot be sufficiently described either. In these works, the performance metrics (e.g., the expected total cost) are computed based on the deterministic characterization of individual workstations (and systems), and may well deviate substantially from the real situations.

2.2 Optimization via Simulation (OvS) Methods

Liu et al. [64] adapted an OvS method to solve the production planning problem for a scaled-down semiconductor fabrication system. The DES can mimic the target manufacturing system with any desired details, and can naturally accommodate any customer demand patterns. Initialized at the current status of the real system of interest, the DES can simulate the manufacturing and demand fulfillment process under a candidate release plan, and obtain the system performance metrics over the planning horizon. With multiple simulation replications, the performance metrics associated with a release plan can be estimated. Built on DES' evaluation ability, OvS can be performed [42]. Although accurately relating the performance metrics to the release decision, performing DES could be very consuming. As the complexity of system increases, this drawback becomes critical and OvS may well not be able to generate good decisions in a reasonable amount of time.

Chapter 3

Methodology Overview

In light of the limitations of the existing methods in the production planning literature, we developed a metamodeling-based approach: The plan optimization is eventually solved in an OvS scheme, whereas the computationally expensive DES is replaced by the metamodel-based MCS, which allows for an accurate and timely quantification of the input-output relationships for manufacturing systems. The development of the metamodel-based MCS is the key contribution of this work, and it serves as the foundation for responsive plan optimization.

Recall the input-output process illustrated in Figure 1.1. For convenience of discussion, the following notations are used:

Δt: the time interval considered as the basic time unit. All the time variables/parameters in this work are measured in terms of the time unit Δt.

t: the time index measured in terms of the basic time unit.

H: the length of planning horizon, with H given in terms of the basic time unit.

$A(t)$: the input release (or arrival) process to the system which counts the number of arrivals during the time interval $(t, t+1]$.

$x(t) = \mathrm{E}[A(t)]$: the first moment of the arrival process $A(t)$.

$Q(t)$: the state process of the system which counts the number of work in process (WIP) in the system at time t.

$m_i(t) = \mathrm{E}[Q^i(t)]$: the i^{th} moment of the state process $Q(t)$ $(i = 1, 2)$.

$m_{1j}(t) = \mathrm{E}[Q(t)Q(t-j)]$: the first moment of $Q(t)Q(t-j)$ $(j = 1, 2, \ldots, J_Q)$, where J_Q is the highest order of the time lag needed to describe the WIP process.

$D(t)$: the departure process from the system which counts the number of finished jobs during the time interval $(t, t+1]$.

$d_i(t) = \mathrm{E}[D^i(t)]$: the i^{th} moment of the departure process $D(t)$ $(i = 1, 2)$.

$d_{1j}(t) = \mathrm{E}[D(t)D(t-j)]$: the first moment of $D(t)D(t-j)$ $(j = 1, 2, \ldots, J_D)$, where J_D is the highest order of the time lag needed to describe the departures.

$e_{QD}(t) = \mathrm{E}[Q(t)D(t)]$: the first moment of $Q(t)D(t)$.

$\mathcal{D}(t)$: the demand process which counts the quantity requested by customers at the end of time t.

The input $A(t)$ is assumed to be completely characterized by $x(t)$, which is a common assumption in all the existing production planning works. For detailed justification of this assumption, please refer to Section 1.4 of Yang and Liu [90].

The production planning task is thus to determine the release plan $\{x(t); t = 1, 2, \ldots, H\}$. The outputs, $\{Q(t), D(t); t = 1, 2, \ldots, H\}$, are bivariate time-series counts described by the characteristics vector

$$\mathbf{y}(t) = (m_1(t), m_2(t), d_1(t), d_2(t), m_{11}(t), \ldots, m_{1J_Q}(t), d_{11}(t), \ldots, d_{1J_D}(t), e_{QD}(t))^\top. \quad (3.1)$$

The demand $\mathcal{D}(t)$ is a given time-series process pre-specified by forecasting methods outside of the scope of production planning. The output processes $\{Q(t), D(t); t = 1, 2, \ldots, H\}$, interact with the demand $\{\mathcal{D}(t); t = 1, 2, \ldots, H\}$ and determine the system performance over $t = 1, 2, ..., H$, which typically includes the total cost and demand fulfill rate.

In search of the optimal release plan, a multi-objective genetic algorithm (MOGA) is adopted to explore the input decision space, as will be seen in Chapter 6. For each candidate plan, its performance metrics are evaluated following the two steps shown in Figure 3.1.

Step (1): For an arbitrary release plan specified by $\{x^*(t); t = 1, 2, \ldots, H\}$, the pre-obtained metamodel is employed to predict the characteristics (3.1) of the output processes $\{Q(t), D(t); t = 1, 2, \ldots, H\}$ over the planning horizon. The prediction can be made in no time and with the high-fidelity of DES, since the metamodel is a mathematical approximation estimated from DES data. As will be detailed in Chapter 4, the metamodel takes the form of difference equations, is fitted from extensive DES data obtained offline (prior to performing plan optimization), and is able to accurately quantify the dependence of the output characteristics (3.1) upon the input release plan.

Metamodeling is to bridge the gap between the time-consuming DES and the need for responsive decision making [4]. Once the configuration of a manufacturing system is established, its DES can be developed and kept running for weeks (or even months) to provide DES data for the estimation of the metamodel. The resulting metamodel not only embodies the high fidelity of DES, but also allows for quick ”what-if” analysis, and hence can be used to support responsive production optimization when the need for decision arises.

Step (2): As pointed out in Chapter 1, the performance metrics result from the interactions of general time-series counts $\{Q(t), D(t); t = 1, 2, \ldots, H\}$ and $\{\mathcal{D}(t); t = 1, 2, \ldots, H\}$, and they cannot be assessed analytically. In this work, an MCS method for time-series counts is developed (in Chapter 5) to simulate $\{Q(t), D(t); t = 1, 2, \ldots, H\}$ based on the output characteristics (3.1) obtained from Step (1), and to numerically evaluate the performance metrics in a timely manner. This evaluation step requires the time of fast MCS, as opposed to the time-consuming DES.

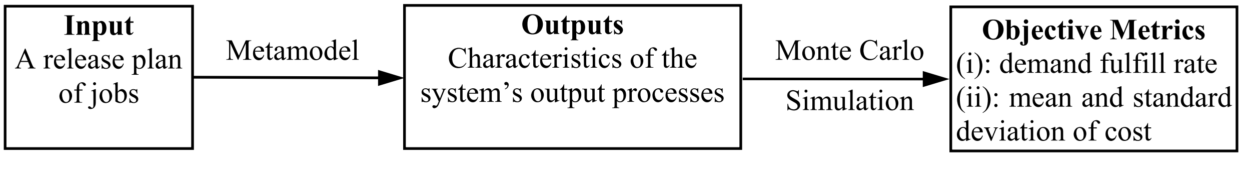


Figure 3.1: Evaluating a release plan's performance.

Hence, the two steps shown in Figure 3.1 allow for the accurate and timely evaluation of a release plan, and hence provide the necessary basis for high-quality and responsive optimization of production plans.

Chapter 4

Metamodeling via Offline Discrete-Event Simulation

The metamodel in Step (1) of Figure 3.1 characterizes the time-dependent behavior of a general manufacturing system, and is obtained by extending the metamodeling method in Yang and Liu [90]. In [90], the metamodel is fitted from DES data to quantify the relationship between $x(t)$, the first moment of the arrivals $A(t)$, and $\{m(t), d(t)\}$, the first-moment measures of the output processes $\{Q(t), D(t)\}$. As noted earlier, $\{Q(t), D(t)\}$ are bivariate time-series counts, and thus their first-moment measures are not adequate to characterize these processes. In this work, the metamodeling in [90] is extended to quantify the functional relationships between the input $x(t)$ and the $\mathbf{y}(t)$ vector, which includes the first- as well as second-moment measures of $\{Q(t), D(t)\}$ as shown in (3.1). The metamodeling methods are described bellow, and emphasis goes to the aspects different from those in [90].

4.1 Functional Form of the Metamodel

As in Yang and Liu [90], the metamodel takes the form of difference equations, and the characteristics vector $\mathbf{y}(t)$ can be expressed in general as

$$\mathbf{y}(t) = \mathbf{F}(x(t-1), x(t-2), ..., \mathbf{y}(t-1), \mathbf{y}(t-2), ...), \tag{4.1}$$

where $\mathbf{F}$ is a vector function with compatible dimension of $\mathbf{y}(t)$. Each component function F_i specifies the dependence of the i^{th} element of $\mathbf{y}(t)$ upon a given input $\{x(t-1), x(t-2), \ldots\}$ and the system's historical outputs $\{\mathbf{y}(t-1), \mathbf{y}(t-2), \ldots\}$. As in [90], F_i assumes the functional form of a third-order polynomial. The metamodel (4.1) is estimated based on the data obtained from running DES of the manufacturing system.

4.2 Sampling via Discrete-Event Simulation (DES)

To fit the metamodel (4.1) that functionally relates $x(t)$ to $\mathbf{y}(t)$, extensive DES experiments are carried out. For the DES, job arrivals are modeled by a time-varying process (e.g., non-stationary Poisson) which is characterized by its arrival rate $x(t)$. Each simulation run is performed by feeding the stochastic arrivals with a pre-specified $x(t)$ to the system for a simulation length of L time units (Δt), and a total of R simulation replications are obtained. The specification of the input $x(t)$, L and R are detailed in [90], and also provided in Appendix B for reader's convenience.

From the r^{th} replication, the arrival, WIP state and departure processes $\{A_r(t)$, $Q_r(t)$, $D_r(t)$; $t = 1, 2, \ldots, L\}$ are recorded. Based on the R replications, the input-output data for the metamodel fitting are denoted as $\{(\widetilde{x}(t), \widetilde{\mathbf{y}}(t)); t = 1, 2, \ldots, L\}$ and calculated as follows:

$$(\widetilde{x}(t), \widetilde{\mathbf{y}}(t))^t = \begin{pmatrix} \widetilde{x}(t) \\ \widetilde{m}_1(t) \\ \widetilde{m}_2(t) \\ \widetilde{d}_1(t) \\ \widetilde{d}_2(t) \\ \widetilde{m}_{11}(t) \\ \vdots \\ \widetilde{m}_{1J_Q}(t) \\ \widetilde{d}_{11}(t) \\ \vdots \\ \widetilde{d}_{1J_D}(t) \\ \widetilde{e}_{QD}(t) \end{pmatrix} = R^{-1} \sum_{r=1}^{R} \begin{pmatrix} A_r(t) \\ Q_r(t) \\ Q_r(t)^2 \\ D_r(t) \\ D_r(t)^2 \\ Q_r(t)Q_r(t-1) \\ \vdots \\ Q_r(t)Q_r(t-J_Q) \\ D_r(t)D_r(t-1) \\ \vdots \\ D_r(t)D_r(t-J_D) \\ Q_r(t)D_r(t) \end{pmatrix}. \tag{4.2}$$

4.3 Model Fitting and Selection

From the data (4.2), the metamodel (4.1) is fitted. The various model selection issues are discussed in details in Section 5.2 of Yang and Liu [90] regarding the functional terms included in $\mathbf{F}$ of the metamodel. The additional model selection issue involved in modeling $\mathbf{y}(t)$, which includes not only the first-moment but also the second-moment measures of the output processes, lies in the determination of the time lags, J_Q and J_D. As defined in Chapter 3, J_Q and J_D are the highest orders of autocorrelation needed to characterize the processes $Q(t)$ and $D(t)$ respectively. Given the DES data $\{Q_r(t), D_r(t); t = 1, 2, \ldots, L; r = 1, 2, \ldots, R\}$, J_Q and J_D can be determined by inspecting the sample partial autocorrelation function (SPACF) [14] of the time series.

The details of model fitting is provided in Appendix C of this work for readers' convenience. In Appendix D, the equations of fitted metamodel are given in (D.1)-(D.9), which describes the non-stationary behavior of a scaled-down semiconductor fabrication system (Appendix A).

4.4 Metamodel-Based Prediction

Once the metamodel has been obtained, it can be used to predict within a second the system's behavior under any input rate $x^*(t)$ over a time horizon. Specifically, suppose that we are currently standing at time 0, where typically the system history $\{\mathbf{y}(t); t \leq 0\}$ is available. Using the historical outputs $\{\mathbf{y}(t); t \leq 0\}$ as the seed values to initiate the metamodel-based computation, the future characteristic processes $\{\widehat{\mathbf{y}}(t); t = 1, 2, \ldots, H\}$ can be recursively predicted via the metamodel for any $\{x^*(t); t = 1, 2, \ldots, H\}$.

It is worth noting that the time to complete the metamodel-based recursive computation for future prediction is not sensitive at all to the complexity of the system being investigated, since the computation is performed based on the fitted metamodel, which only involves the basic calculations such as additions and multiplications.

Chapter 5

Metamodel-Based Monte Carlo Simulation (MCS) for System Output Processes

This chapter is concerned with Step (2) of Figure 3.1: For a release plan $\{x^*(t); t = 1, 2, \ldots, H\}$, how to evaluate the resulting system performance, having obtained the output characteristics $\{\widehat{\mathbf{y}}(t); t = 1, 2, \ldots, H\}$ from the metamodel prediction.

As explained in Chapter 1, the performance metrics result from the interactions of the time series $\{Q(t), D(t); t = 1, 2, \ldots, H\}$ and the customer demand $\{\mathcal{D}(t); t = 1, 2, \ldots, H\}$ throughout the planing horizon, and cannot in general be evaluated analytically. Hence, an MCS method is developed in this work to bridge $\{\mathbf{y}(t); t = 1, 2, \ldots, H\}$ and the performance metrics. The MCS is able to quickly generate time-series $\{\widehat{Q}(t), \widehat{D}(t); t = 1, 2, \ldots, H\}$, whose major characteristics match the metamodel-predicted $\{\widehat{\mathbf{y}}(t); t = 1, 2, \ldots, H\}$. Clearly, the MCS-simulated processes $\{\widehat{Q}(t), \widehat{D}(t); t = 1, 2, \ldots, H\}$ are meant to mimic real (or DES) output processes $\{Q(t), D(t); t = 1, 2, \ldots, H\}$.

In this work, a parametric approach is taken to generate $\{\widehat{Q}(t), \widehat{D}(t); t = 1, 2, \ldots, H\}$ via MCS: First, a parametric model family of the output time series is identified (Section 5.1); Second, with the selected model family, the time-series model is fitted which possesses the metamodel-predicted characteristics $\{\widehat{\mathbf{y}}(t); t = 1, 2, \ldots, H\}$ (Section 5.2.1); Third, based on the fitted time-series model, MCS is carried out to generate $\{\widehat{Q}(t), \widehat{D}(t); t = 1, 2, \ldots, H\}$ by employing certain pseudo-random generation mechanisms (Section 5.2.2).

5.1 Time-Series Model Identification

The model identification for the output time series is performed offline based on the DES data obtained for metamodeling. To establish the appropriate model for the bivariate process $(Q(t), D(t))^\top$, the first task is to identify the univariate time-series models suitable to describe $Q(t)$ and $D(t)$ respectively (Section 5.1.1). Then, the univariate models are combined to form a bivariate time-series model for the joint processes $(Q(t), D(t))^\top$, by recognizing the correlation between $Q(t)$ and $D(t)$ (Section 5.1.3).

5.1.1 Candidate Models of Univariate Time Series

For convenience of discussion, denote $Z(t)$ as a univariate time series representing an output process (that is, $Q(t)$ or $D(t)$) from the system. What family of univariate time-series models is suitable to describe $Z(t)$? First, $Z(t)$ is a counting process over time, and it takes non-negative integer values. Second, such a process from a general queueing system (e.g., manufacturing systems) could be over-dispersed, equi-dispersed or under-dispersed, which means that the variance/mean ratio of $Z(t)$ could be greater, equal or less than 1 [27]. Third, the autocorrelations of $Z(t)$ could involve positive and/or negative values. Fourth, the selected model family is also required to render a straight-forward relationship between its model parameters and the process characteristics $\widehat{\mathbf{y}}(t)$, since the time-series model herein serves as a vehicle to generate MCS data $\{\widehat{Q}(t), \widehat{D}(t); t = 1, 2, \ldots, H\}$ that possess the given characteristics $\widehat{\mathbf{y}}(t)$.

In light of the above-mentioned model requirements, we have explored the current time-series literature [50, 51, 88, 28, 35, 68, 49, 37, 55, 56, 3, 87, 47, 25, 92, 93, 94, 61], and adapted/adopted the existing models to obtain three non-stationary time-series models as the potential candidates for $Z(t)$: N-AR, N-INAR and N-SINAR models. Each of these three models represents the non-stationary extension from its stationary counterpart by allowing the model parameters to vary with time. All the candidates are autoregressive models, and can be written in the general autoregressive form:

$$Z(t) = \sum_{j=1}^{J} \mathcal{R}(\alpha_j(t), Z(t-j)) + \varepsilon(t); \qquad t = 1, 2, \ldots \tag{5.1}$$

In (5.1), J is a positive integer representing the order of the autoregressive model, $\varepsilon(t)$ denotes an independently-distributed innovation, $\{\alpha_j(t); j = 1, 2, \ldots, J; t = 1, 2, \ldots\}$ are time-dependent model parameters, and $\mathcal{R}$ represents a random operator. It is assumed that all the random operations in $\mathcal{R}$ are performed independently of each other and also independent of the innovation process $\{\varepsilon(t); t = 1, 2, \ldots\}$.

The properties of these three models are summarized in Table 5.1. In the order of increasing complexity, N-AR, N-INAR and N-SINAR models are described as follows.

Table 5.1: Summary of the three univariate time-series models.

	Distribution	Feasible range	Able to accommodate negative	Feasible range
	of $\varepsilon(t)$	of $Z(t)$	autocorrelations?	of $\alpha_j(t)$
N-AR	Normal	$\mathbb{R}$	Yes	$(-\inf, \inf)$
N-INAR	GP	$\mathbb{N}$	No	$[0, 1]$
N-SINAR	GP	$\mathbb{Z}$	Yes	$[-1, 1]$

Model 1: N-AR model is extended from the classic AR models [14] to have time-varying parameters $\{\alpha_j(t); j = 1, 2, \ldots, J; t = 1, 2, \ldots\}$ [33, 34, 76, 74, 20, 10]. For N-AR, the general model (5.1) takes the specific form of

$$Z(t) = \sum_{j=1}^{J} \alpha_j(t) Z(t-j) + \varepsilon(t); \qquad t = 1, 2, \ldots, \tag{5.2}$$

where $Z(t) \in \mathbb{R}$, and $\varepsilon(t)$ is independently normally distributed.

Model 2: N-INAR model is extended from the integer valued autoregressive (INAR) model [28, 50, 88] by using time-varying parameters [15, 30]. For N-INAR, the general model (5.1) takes the specific form of

$$Z(t) = \sum_{j=1}^{J} \alpha_j(t) \circ Z(t-j) + \varepsilon(t); \qquad t = 1, 2, \ldots, \tag{5.3}$$

where $Z(t) \in \mathbb{N}$. The innovation $\varepsilon(t)$ is an independent non-negative integer-valued random variable, and controls the dispersion behavior of $Z(t)$ [88]. In order for N-INAR to model over-, equi- and under-dispersed data, the following discrete distributions could be used for $\varepsilon(t)$: generalized Poisson (GP) [24], double Poisson [29], COM-Poisson [80], Lerch distribution [88], or weighted Poisson [88]. In this work, GP is adopted for $\varepsilon(t)$ out of the following considerations: First, assuming GP for $\varepsilon(t)$ allows the distribution parameters of $\varepsilon(t)$ to be straight-forwardly derived from the data characteristics $\widehat{\mathbf{y}}(t)$; Second, GP is most widely studied and used in modeling integer-valued time series [94, 88].

The operator "$\circ$" in (5.3) denotes the binomial thinning process [84] which is defined as

$$\alpha \circ n = \sum_{m=1}^{n} C_m \text{ for } n \geq 0 \tag{5.4}$$

where $\alpha \in [0, 1]$ is the operation parameter and C_m is a sequence of independent and identically distributed Bernoulli random variables independent of $Z(t)$, with $\mathbb{P}(C_m = 1) = \alpha$. Since α takes the value of $\alpha_j(t)$ $(j = 1, 2, \ldots, J)$ in (5.3), $\alpha_j(t)$ is restricted to be within the range of $[0, 1]$; and consequently, N-INAR only allows for time series with certain positive autocorrelation patterns [14].

Model 3: N-SINAR N-SINAR is the non-stationary extension of the signed integer-valued autoregressive (SINAR) model introduced in [55] and [23]. For N-SINAR, the general model (5.1) takes the specific form of

$$Z(t) = \sum_{j=1}^{J} F(\alpha_j(t)) \circ Z(t-i) + \varepsilon(t), \tag{5.5}$$

where $Z(t) \in \mathbb{Z}$. The innovation $\varepsilon(t)$ is an independent integer-valued random variable. As with N-INAR, the GP distribution is employed for $\varepsilon(t)$ in N-SINAR, and allows N-SINAR to handle over-, equi- and under-dispersed data.

The operator "$F(\cdot)\circ$" denotes the generalized thinning process [55]

$$F(\alpha) \circ n = \begin{cases} \operatorname{sign}(n) \sum_{m=1}^{|n|} C_m & \text{if } n \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

where $\alpha \in [-1, 1]$ is the operation parameter; $\operatorname{sign}(n)$ equals to 1 if $n \geq 0$ and -1 if $n < 0$; and $\{C_m; m = 1, 2, \ldots\}$ are independent and identically distributed random variables with the probability distribution given as below [23]

$$\mathbb{P}(C_m = -1) = \Big(\frac{1-\alpha}{2}\Big)^2, \;\; \mathbb{P}(C_m = 0) = \frac{1-\alpha^2}{2}, \;\; \mathbb{P}(C_m = 1) = \Big(\frac{1+\alpha}{2}\Big)^2. \qquad (5.6)$$

Since α takes the value of $\alpha_j(t)$ ($j = 1, 2, \ldots, J$) in (5.5), $\alpha_j(t)$ is restricted to be within the range of $[-1, 1]$; and thus, N-SINAR is able to accommodate both positive and negative serial correlations due to its flexible thinning operators (5.6).

5.1.2 Model Selection for Univariate Time Series

From extensive offline DES (Section 4.2), the time-series data for an output process can be obtained and represented as $\{Z_r(t); t = 1, 2, \ldots, L; r = 1, 2, \ldots, R\}$. Recall that L denotes the simulation length of each replication, and R the number of replications. Clearly, $Z_r(t)$ could be the WIP state $Q_r(t)$ or the departure $D_r(t)$ as denoted in Section 4.2. As discussed in 4.3, the autoregressive order J in model (5.1) can be determined based on the data $\{Z_r(t); t = 1, 2, \ldots, L; r = 1, 2, \ldots, R\}$.

The remaining question yet to be answered is: How to select from the three-model ensemble (i.e., N-AR, N-INAR, and N-SINAR) the most appropriate one to describe the data? Both transient and steady-state data are included in $\{Z_r(t); t = 1, 2, \ldots, L; r = 1, 2, \ldots, R\}$ following the design of DES experiments in [90]. For the purpose of time-series model selection, only the steady-state data are used for convenience. In the spirit of finding the simplest and the most adequate model, we provide the following guidelines.

First, the steady-state data are subjected to normality tests, such as QQ-plot [89], Kolmogorov-Smirnov test [67] or Shapiro-Wilk test [82], to determine if N-AR is appropriate. Although the time series are non-negative integer counts, the simplest N-AR may provide an adequate fit, especially when the counts are large. If the data fail the normality tests, then the integer-valued models (that is, N-INAR and N-SINAR) will be considered in the follow-up step.

Second, as the relatively simple model, N-INAR is favored over N-SINAR unless it is inadequate to capture the important data features. As shown in Table 5.1, N-INAR and N-SINAR differ in terms of the feasible range of the parameters $\{\alpha_j(t); j = 1, 2, \ldots, J\}$, which lead to their different ranges of allowed autocorrelation patterns. Hence, the estimation and inference of $\{\alpha_j; j = 1, 2, \ldots, J\}$ are used to guide the model selection between INAR and SINAR. More specifically, from the stationary DES data, the sampling autocorrelation function (SACF) can be calculated and fed to the Yule-Walker equation [16, 14] to obtain the estimates of $\{\alpha_j; j = 1, 2, \ldots, J\}$. Inferences on the Yule-Walker estimates can also be derived, and the following hypothesis test can be performed:

$$H_0 : \alpha_j \geq 0; \quad vs. \quad H_a : \alpha_j < 0 \tag{5.7}$$

for $j = 1, 2, ..., J$. If H_0 is rejected for any j, N-SINAR will be selected; otherwise, N-INAR will be adopted. The details for the Yule-Walker estimation and inference are given in Appendix E.

5.1.3 Bivariate Time-Series Model

Following the model-selection procedure (Section 5.1.2), two univariate time-series models are selected for the two processes, $Q(t)$ and $D(t)$, respectively, and can then be used to form the bivariate time-series model for $(Q(t), D(t))^\top$.

The bivariate time series $(Q(t), D(t))^\top$ is modeled as:

$$\begin{aligned} Q(t) &= \sum_{j=1}^{J_Q} \mathcal{R}^{(Q)}(\alpha_j^{(Q)}(t), Q(t-j)) + \varepsilon^{(Q)}(t) && (5.8) \\ D(t) &= \sum_{j=1}^{J_D} \mathcal{R}^{(D)}(\alpha_j^{(D)}(t), D(t-j)) + \varepsilon^{(D)}(t) && (5.9) \end{aligned}$$

where $\boldsymbol{\varepsilon}(t) = (\varepsilon^{(Q)}(t), \varepsilon^{(D)}(t))^\top$ is the innovation vector with

$$\mathrm{Var}[\boldsymbol{\varepsilon}(t)] = \begin{pmatrix} \mathrm{Var}[\varepsilon^{(Q)}(t)] & \mathrm{Cov}[\varepsilon^{(Q)}(t), \varepsilon^{(D)}(t)] \\ \mathrm{Cov}[\varepsilon^{(Q)}(t), \varepsilon^{(D)}(t)] & \mathrm{Var}[\varepsilon^{(D)}(t)] \end{pmatrix}. \tag{5.10}$$

In the bivariate model, the marginal distribution of $\varepsilon^{(Q)}(t)$ or $\varepsilon^{(D)}(t)$ is specified by the corresponding univariate time-series model (Section 5.1.1); J_Q and J_D are the orders of the univariate autoregressive models respectively; the specific forms of the random operators, $\mathcal{R}^{(Q)}$ and $\mathcal{R}^{(D)}$, correspond to the selected univariate models for $Q(t)$ and $D(t)$ respectively.

5.2 Metamodel-Based Monte Carlo Simulation

As can be seen from Section 5.1, a bivariate time-series model family (5.8–5.9) can be selected for the target processes $(Q(t), D(t))^\top$ based on offline DES. The selected model can then be fitted from the given characteristics $\widehat{\mathbf{y}}(t)$ (Section 5.2.1). The fitted bivariate model allows for the MCS of $(\widehat{Q}(t), \widehat{D}(t))^\top$ (Section 5.2.2).

5.2.1 Time-Series Model Fitting

Denote the fitted bivariate time-series model as

$$\begin{aligned} \widehat{Q}(t) &= \sum_{j=1}^{J_Q} \mathcal{R}^{(Q)}(\widehat{\alpha}_j^{(Q)}(t), \widehat{Q}(t-j)) + \widehat{\varepsilon}^{(Q)}(t) && (5.11) \\ \widehat{D}(t) &= \sum_{j=1}^{J_D} \mathcal{R}^{(D)}(\widehat{\alpha}_j^{(D)}(t), \widehat{D}(t-j)) + \widehat{\varepsilon}^{(D)}(t). && (5.12) \end{aligned}$$

The fitted parameters include $\{\widehat{\alpha}_j^{(Q)}(t); j = 1, 2, \ldots, J_Q\}$, $\{\widehat{\alpha}_j^{(D)}(t); j = 1, 2, \ldots, J_D\}$, $\widehat{\mathrm{E}}[\widehat{\varepsilon}^{(Q)}(t)]$, $\widehat{\mathrm{E}}[\widehat{\varepsilon}^{(D)}(t)]$, $\widehat{\mathrm{Var}}[\widehat{\varepsilon}^{(Q)}(t)]$, $\widehat{\mathrm{Var}}[\widehat{\varepsilon}^{(D)}(t)]$ and $\widehat{\mathrm{Cov}}[\widehat{\varepsilon}^{(Q)}(t), \widehat{\varepsilon}^{(D)}(t)]$, which can be derived analytically as functions of the metamodel-predicted characteristics $\widehat{\mathbf{y}}(t)$. Please refer to Appendix F for the derivation.

Hence, the bivariate process described by the fitted model possesses the same characteristics as those specified by $\widehat{\mathbf{y}}(t)$. Specifically,

- The marginal mean, variance, and lag-j ($j = 1, 2, \ldots, J_Q$) autocorrelations of $\widehat{Q}(t)$ are the same as those specified by $\widehat{\mathbf{y}}(t)$; the same can be concluded for $\widehat{D}(t)$.
- The covariance of $\widehat{Q}(t)$ and $\widehat{D}(t)$ is the same as that specified by $\widehat{\mathbf{y}}(t)$.

5.2.2 Monte Carlo Simulation of the System Output Processes

The fitted bivariate model (5.11)–(5.12) can be used to perform MCS for the generation of $(\widehat{Q}(t), \widehat{D}(t))^\top$ time series mimicking the real WIP-state and departure processes.

The historical data $\{Q(t), D(t); t = 0, -1, ..., -\max(J_Q, J_D)+1\}$ is typically available to serve as the seed to initiate the computations (5.11)-(5.12). With given mean vector $(\widehat{\mathrm{E}}[\widehat{\varepsilon}^{(Q)}(t)], \widehat{\mathrm{E}}[\widehat{\varepsilon}^{(D)}(t)])^\top$ and variance–covariance matrix

$$\widehat{\mathrm{Var}}[\boldsymbol{\varepsilon}(t)] = \begin{pmatrix} \widehat{\mathrm{Var}}[\widehat{\varepsilon}^{(Q)}(t)] & \widehat{\mathrm{Cov}}[\widehat{\varepsilon}^{(Q)}(t), \widehat{\varepsilon}^{(D)}(t)] \\ \widehat{\mathrm{Cov}}[\widehat{\varepsilon}^{(Q)}(t), \widehat{\varepsilon}^{(D)}(t)] & \widehat{\mathrm{Var}}[\widehat{\varepsilon}^{(D)}(t)] \end{pmatrix}, \tag{5.13}$$

the bivariate innovation process $(\widehat{\varepsilon}^{(Q)}(t), \widehat{\varepsilon}^{(D)}(t))^\top$ can be simulated. As explained in Section 5.1.1, herein a univariate innovation follows a continuous normal or discrete GP distribution. If both $\widehat{\varepsilon}^{(Q)}(t)$ and $\widehat{\varepsilon}^{(D)}(t)$ are normally distributed, the bivariate innovations can be easily generated by the simulation algorithms for multivariate normal distributions [79]. If both innovation processes follow discrete GP, the copula-based algorithm in Avramidis et al. [9] can be used for simulation. Otherwise, if one is discrete and the other is continuous, then the algorithm in Channouf and L'Ecuyer [21] can be employed. The copula-based algorithms developed by [9, 21] are given in Appendix G for readers' convenience.

Chapter 6

Metamodel-Based Optimization for Online Production Planning

Optimization of production planning is performed by utilizing the fast and high-fidelity evaluation ability rendered by the metamodel-based MCS, which relates the input release plan to the system's performance over a planning horizon.

6.1 Problem Formulation of Production Planning

The plan optimization problem is formulated following the basic structure in the existing production literature [64, 70, 54, 52, 8, 7]. The problem formulation also represents an extension for generality, which is allowed by our metamodel-based methods.

Following the notations in Chapter 3, the planning horizon is denoted as $(0, H]$ and divided into P equal-length time buckets. The H is given in terms of the number of time units Δt. The time length of each bucket is set based on the practical needs, and could be as short as one time unit. Within each bucket, the job release rate is assumed to be constant and equals to x_p with $p = 1, 2, ..., P$. Thus, the release plan over $(0, H]$ can be specified by a $P \times 1$ vector

$$\mathbf{x} = (x_1, x_2, ..., x_P)^\top,$$

which is the vector of decision variables for plan optimization. The customer demand is a given stochastic time series provided by forecasting efforts outside of the scope of production planning. As in Chapter 3, the demand is denoted as $\{\mathcal{D}(t); t = 1, 2, \ldots, H\}$. Depending on when demand is actually fulfilled in real processes, $\mathcal{D}(t)$ could be set as 0 at certain time points.

The parameters, dependent variables, and objective criteria of the optimization problem are provided below.

Parameters

w_t the unit holding cost of the WIP per time unit at time t.

h_t the unit holding cost of finished jobs (or products) per time unit at time t.

G_0 the initial number of finished jobs or backlogged orders at time 0, the beginning of the planning horizon.

Dependent Variables

- The output processes from the system, which have been defined in Chapter 3, are rewritten here to stress their dependence on $\mathbf{x}$:

$$\{Q(t, \mathbf{x}), D(t, \mathbf{x}); t = 1, 2, \ldots, H\}. \tag{6.1}$$

Accordingly, the characteristics of (6.1) are denoted as:

$$\begin{aligned}\mathbf{y}(t, \mathbf{x}) = (&m_1(t, \mathbf{x}), m_2(t, \mathbf{x}), d_1(t, \mathbf{x}), d_2(t, \mathbf{x}), m_{11}(t, \mathbf{x}), \ldots, m_{1J_Q}(t, \mathbf{x}) \\ &d_{11}(t, \mathbf{x}), \ldots, d_{1J_D}(t, \mathbf{x}), e_{QD}(t, \mathbf{x}))^\top.\end{aligned} \tag{6.2}$$

whose $\mathbf{x}$-free correspondence is defined in Chapter 3.

- $G(t, \mathbf{x})$: The number of finished jobs at the end of the t^{th} time unit, after the inventory has been refilled by the newly-completed jobs and the demand (if any) has been realized in that time unit. It is calculated as

$$G(t, \mathbf{x}) = G(t-1, \mathbf{x}) + D(t, \mathbf{x}) - \mathcal{D}(t); \;\; t = 1, 2, \ldots, H, \tag{6.3}$$

and is a resulting time series due to the interaction of the two stochastic time series $\{D(t, \mathbf{x}); t = 1, 2, \ldots, H\}$ and $\{\mathcal{D}(t); t = 1, 2, \ldots, H\}$. In this formulation, $G(t, \mathbf{x})$ is allowed to be a negative integer, which represents the backlogged shortage by the end of the the t^{th} time unit.

- $I(t, \mathbf{x})$: The actual inventory of finished jobs at the end of the t^{th} time unit, which is given as:

$$I(t, \mathbf{x}) = \max\{G(t, \mathbf{x}), 0\}; \quad t = 1, 2, \ldots, H. \tag{6.4}$$

- $B(t, \mathbf{x})$: The amount of customer demand that are realized in the t^{th} time unit but cannot be satisfied immediately with the existing inventory. These demands will be backlogged and fulfilled based on first-come-first-serve.

$$B(t, \mathbf{x}) = \begin{cases} \mathcal{D}(t) & \text{if } G(t-1, \mathbf{x}) + D(t, \mathbf{x}) < 0 \\ -\min\{G(t-1, \mathbf{x}) + D(t, \mathbf{x}) - \mathcal{D}(t), 0\} & \text{otherwise} \end{cases} \tag{6.5}$$

where $t = 1, 2, ..., H$.

Objective Criteria

The total cost associated with a release plan $\mathbf{x}$ can be written as:

$$TC(\mathbf{x}) = \sum_{t=1}^{H} w_t Q(t, \mathbf{x}) + \sum_{t=1}^{H} h_t I(t, \mathbf{x}) \tag{6.6}$$

including two types of cost: (i) the WIP holding cost, and (ii) the finished-goods inventory cost. The total cost $TC(\mathbf{x})$ is a random variable dependending on the release plan $\mathbf{x}$, and it is of interest to quantify both $\mathrm{E}[TC(\mathbf{x})]$, the expected cost, and $\mathrm{Std}[TC(\mathbf{x})]$, the standard deviation of the cost.

The backlogged shortage $\{B(t, \mathbf{x}); t = 1, 2, \ldots, H\}$ can be used to obtain the demand fulfill rate $DF(\mathbf{x})$, the percentage of customer demand that can be fulfilled immediately with existing inventory:

$$DF(\mathbf{x}) = \frac{\sum_{t=1}^{H} B(t, \mathbf{x})}{\sum_{t=1}^{H} \mathcal{D}(t)} \tag{6.7}$$

The objectives of production planning could be to minimize $\mathrm{E}[TC(\mathbf{x})]$ and/or to maximize $\mathrm{E}[DF(\mathbf{x})]$.

It is worth mentioning that the formulation above represents one of the possible ways to form the optimization problem of production planning [71]. Our metamodel-based MCS method can accommodate any formulation, depending on the practical needs.

6.2 Solving the Optimization Problem

Figure 6.1 outlines the procedure used to search for the production plans that consider the cost and service performance. This online production planning is performed with the two obtained offline efforts: (a) the metamodel relating a release plan $\mathbf{x}$ to $\{\widehat{\mathbf{y}}(t, \mathbf{x}); t = 1, 2, \ldots, H\}$, the characteristics of the output processes $\{Q(t), D(t); t = 1, 2, \ldots, H\}$; (b) the appropriate bivariate time-series model selected based on the offline DES data.

To consider multiple objective criteria, a multi-objective optimization algorithm such as gamultiobj in Matlab Optimization Toolbox is adopted to search for the Pareto optimal solutions (release plans). Such an algorithm is employed to explore the decision space of $\mathbf{x}$, and to generate candidate plans for the optimization problem, as shown in Step (v) of Figure 6.1.

Each candidate plan $\mathbf{x}$ is evaluated with high-fidelity and in a timely manner, through Steps (i)-(iv). For an $\mathbf{x}$, Step (i) can typically be completed within a second to obtain $\{\widehat{\mathbf{y}}(t, \mathbf{x}); t = 1, 2, \ldots, H\}$ over the planning horizon. Using $\{\widehat{\mathbf{y}}(t, \mathbf{x}); t = 1, 2, \ldots, H\}$, Step (ii) is to obtain the fitted bivariate time-series model for $(Q(t), D(t))^\top$.

In Step (iii), MCS is performed: Based on the fitted time-series model, $\{\widehat{Q}_r(t), \widehat{D}_r(t); t = 1, 2, \ldots, H; r = 1, 2, \ldots, R\}$ is generated, with R being the number of MCS replications; and by using the demand forecasting model, the demand series $\{\widehat{\mathcal{D}}_r(t); r = 1, 2, \ldots, R\}$ is also simulated. The number of replications R required for each $\mathbf{x}$ to obtain estimates of $\mathrm{E}[TC(\mathbf{x})]$, $\mathrm{Std}[TC(\mathbf{x})]$, and $\mathrm{E}[DF(\mathbf{x})]$ with desired statistical precision is determined by a two-stage procedure [64, 90] given in Appendix H.

In Step (iv), the performance metrics $\widehat{\mathrm{E}}[TC(\mathbf{x})]$, $\widehat{\mathrm{Std}}[TC(\mathbf{x})]$ and $\widehat{\mathrm{E}}[DF(\mathbf{x})]$ can be obtained from the multiple replications of MCS data $\{\widehat{Q}_r(t), \widehat{D}_r(t); \widehat{\mathcal{D}}_r(t); t = 1, 2, \ldots, H; r = 1, 2, \ldots, R\}$.

On a computer with Inter(R) Core(TM) i7 CPU and 8G RAM, it takes 0.1 second on average to perform one MCS replication, while one DES replication for the sample system in Chapter 7 takes about 1 second. It is worth pointing out that the DES time depends on the complexity of the real system of interest, whereas the MCS time per replication remains approximately unchanged regardless of the system complexity.

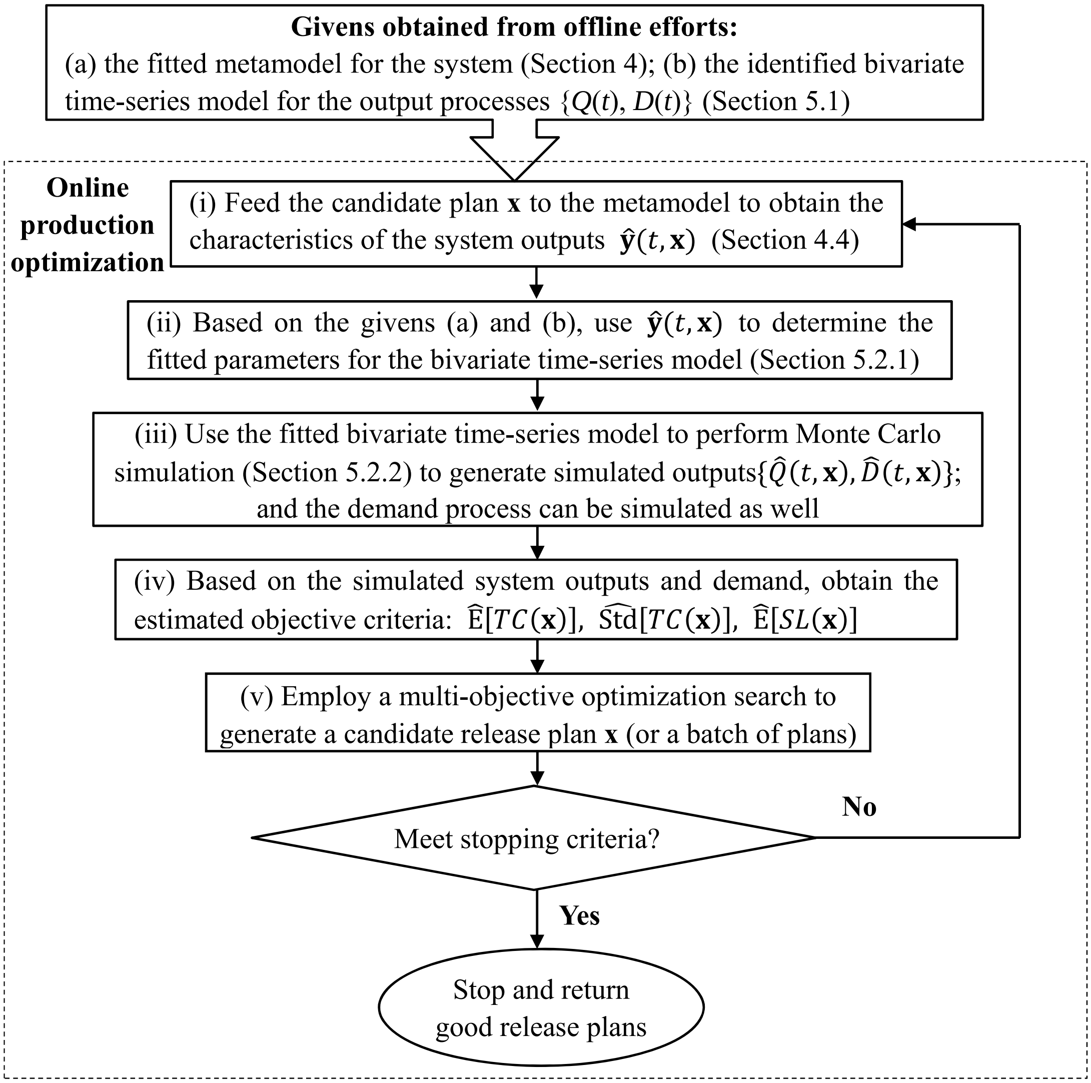

Givens obtained from offline efforts:
(a) the fitted metamodel for the system (Section 4); (b) the identified bivariate time-series model for the output processes {$Q(t)$, $D(t)$} (Section 5.1)
Online production optimization
(i) Feed the candidate plan **x** to the metamodel to obtain the characteristics of the system outputs $\hat{\mathbf{y}}(t, \mathbf{x})$ (Section 4.4)
(ii) Based on the givens (a) and (b), use $\hat{\mathbf{y}}(t, \mathbf{x})$ to determine the fitted parameters for the bivariate time-series model (Section 5.2.1)
(iii) Use the fitted bivariate time-series model to perform Monte Carlo simulation (Section 5.2.2) to generate simulated outputs$\{\hat{Q}(t, \mathbf{x}), \hat{D}(t, \mathbf{x})\}$; and the demand process can be simulated as well
(iv) Based on the simulated system outputs and demand, obtain the estimated objective criteria: $\widehat{\mathrm{E}}[TC(\mathbf{x})]$, $\widehat{\mathrm{Std}}[TC(\mathbf{x})]$, $\widehat{\mathrm{E}}[SL(\mathbf{x})]$
(v) Employ a multi-objective optimization search to generate a candidate release plan **x** (or a batch of plans)
Meet stopping criteria?
No
Yes
Stop and return good release plans

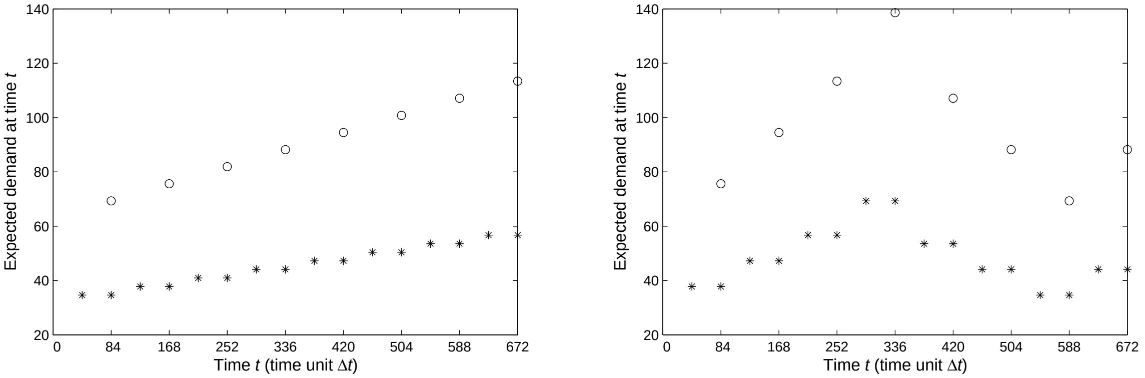
Figure 6.1: Online optimization for production planning.

Chapter 7

Empirical Results

For the purpose of demonstration, the developed method has been applied to solve the production planning problem for a scaled-down semiconductor wafer fabrication system, which was developed in Kayton et al. [58] and detailed in Appendix A. The system consists of 9 workstations, and involves re-entrant flows, machine failures and batch processing, which are the main features of a real semiconductor fabrication system. One type of job is considered and the job arrivals follow Poisson distribution with time-varying arrival rates specified by a release plan. Each job has 14 processing steps and needs to visit some workstations more than once. The raw processing time (not including the waiting time) for a job is expected to be 12.9 hours, and the basic time unit Δt is set as 2 hours.

The production planning problem formulated in Section 6.1 is specified as follows for this case. The length of the planning horizon H is set as 672 time units (eight weeks) and divided into eight equal-length time buckets with each one being 84 time units (one week) long. The cost parameters are assumed to be time-independent and set as: $w_t = 1$ and $h_t = 2$, following the similar cost ratio used in [63, 90]. The initial number of finished jobs G_0 equals to zero. The customer demand $\{\mathcal{D}(t); t = 1, 2, \ldots, H\}$ is a forecasted time series, which in this case is modeled as an AR-GARCH process, a widely-used time-series model for demand forecasting [36, 83, 85, 86]. The simulation algorithm for an AR-GARCH is given in Appendix I. In this study, four AR-GARCH processes of demand are considered, with the expected values and realization times of demand following respectively the four scenarios in Figure 7.1. In Scenario 1, the expected weekly demand increases steadily over the planning



(a) Scenarios 1 and 2

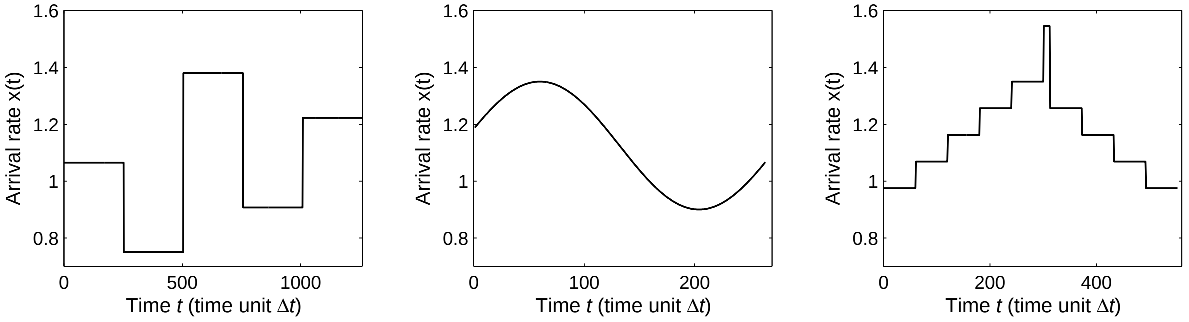
(b) Scenarios 3 and 4

Figure 7.1: Expected demand over the planning horizon of 672 time units (i.e., 8 weeks): "$\circ$" corresponds to Scenarios 1 or 3; "$*$" corresponds to Scenarios 2 or 4.

horizon, and the customer demand is realized once a week; Scenario 2 also follows the general increasing pattern, but allows the demand to be realized twice a week. Scenarios 3 and 4 also correspond to different frequencies of demand realization (as Scenarios 1 and 2 do); their expected demands are fluctuating over the time. The values of expected weekly demand for each scenario are given later in Table 7.2. It is worth noting that in Scenarios 3 and 4, the average demand rate in the 4^{th} week exceeds the system capacity, the maximum rate of production.

7.1 Offline Simulation and Modeling Efforts

Based on the DES data obtained offline, the metamodel was fitted and evaluated (Chapter 4) with the autoregressive order J_Q and J_D both determined as 2. First, DES was carried out to collect the estimation data set (EDS) to fit the metamodel. Following the experimental design strategies in [90], the input arrival rate function $x(t)$ is specified as in Figure 7.2a for the EDS. The $x(t)$ is a piecewise constant function with 5 distinct levels corresponding to five system utilizations: [0.71, 0.50, 0.92, 0.61, 0.82]. The length of a DES replication was set as $L = 1260$ time units with a total of $R = 75000$ replications. The collected EDS takes



(a) EDS (b) VDS 1 (c) VDS 2

Figure 7.2: Input arrival rate functions for the estimation data set (EDS) and two validation data sets (VDS).

the form of (4.2) and is used to fit the metamodel following Section 5.1 of [90]. The fitted metamodel is given in (D.1)–(D.9) of Appendix D.

To evaluate the prediction accuracy of the fitted metamodel, two validation data sets (VDS) were collected by running 70000 replications via DES using the input arrival rate functions shown in Figures 7.2b and 7.2c, respectively. For VDS 1, $x(t)$ is a piecewise constant function which allows the system to be temporarily overloaded. For VDS 2, $x(t)$ is a sine wave function with the highest and lowest arrival rates correspond to system utilizations of 0.9 and 0.6 respectively. These two VDS are designed to test the metamodel's ability to predict both transient and steady-state behavior of the system. The characteristics of system outputs estimated from the VDS serve as the "true" values and are compared with the metamodel-predicted results. As pointed out in Section 4.4, the system's future outputs can be recursively computed by feeding the arrival rate (of a VDS) to the metamodel, and obtained within a second. The comparison results are shown in Figures D.1–D.2 of Appendix D. Evidently, the metamodel prediction, which is represented by the solid curves, is able to accurately track the "true" values, which are plotted as the dashed curves.

After obtaining the offline DES data, the N-AR and N-INAR were identified as the univariate time-series models for $Q(t, \mathbf{x})$ and $D(t, \mathbf{x})$ respectively following Section 5.1. The fitted metamodel and identified time-series models serve as the givens to online production optimization, as shown previously in Figure 6.1.

7.2 Evaluation of Release Plans

For online production optimization, a candidate plan needs to be evaluated following Steps (i)-(iv) of Figure 6.1, by employing the metamodel-based MCS method. Herein, the evaluation quality of the MCS is assessed in comparison with the extensive DES results. An extremely large number of DES replications were performed for each release plan to evaluate the performance metrics with very high precision; the DES estimates are considered as "true" values for the performance metrics and used to assess the accuracy of the MCS estimates.

Table 7.1 provides, for a release plan, the estimates of expected total cost $\mathrm{E}[TC(\mathbf{x})]$, standard deviation of total cost $\mathrm{Std}[TC(\mathbf{x})]$ and expected demand fulfill rate $\mathrm{E}[DF(\mathbf{x})]$, obtained via the MCS method and DES respectively. The deviations of the MCS estimates from the DES ones are also given in the table. A total of 12 release plans are considered, which are derived from the four demand scenarios (Figure 7.1). For instance, the first plan in Table 7.1 is denoted as "Scenario 1×1.05", which means that the expected number of jobs weekly released under this plan equals to 1.05 times the expected weekly demand of Scenario 1. Judging from the relative deviations in Table 7.1, it is clear that the MCS method is able to provide highly accurate estimates for the performance metrics of interest.

Table 7.1: Evaluation quality of the metamodel-based Monte Carlo Simulation (MCS).

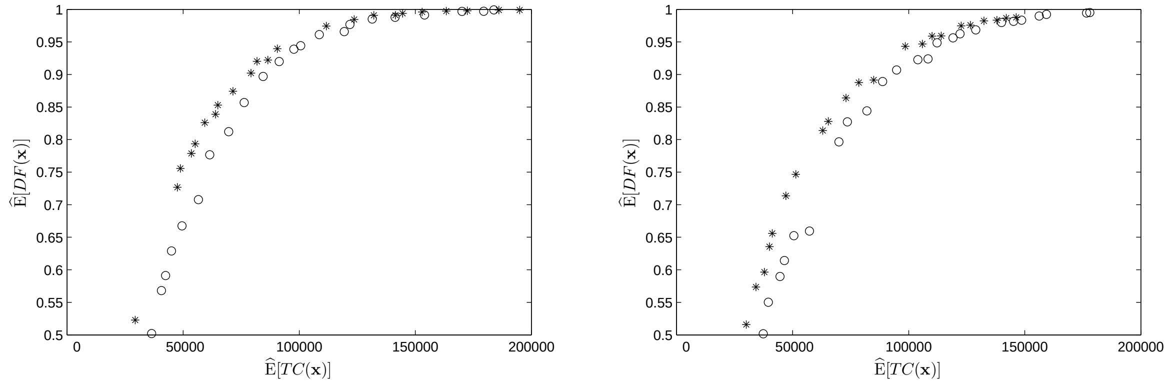
	$\widehat{\mathrm{E}}[TC(\mathbf{x})]$			$\widehat{\mathrm{Std}}[TC(\mathbf{x})]$			$\widehat{\mathrm{E}}[DF(\mathbf{x})]$		
Release plan $\mathbf{x}$	MCS	DES	Deviation	MCS	DES	Deviation	MCS	DES	Deviation
Scenario 1×1.05	$95,369$	$95,037$	0.35%	$35,597$	$33,211$	7.18%	92.77%	93.59%	−0.88%
Scenario 1×1	$74,153$	$74,666$	−0.69%	$31,667$	$30,522$	3.75%	85.30%	86.70%	−1.61%
Scenario 1×0.95	$60,643$	$59,546$	1.84%	$27,895$	$26,182$	6.54%	77.55%	77.26%	0.38%
Scenario 2×1.05	$65,306$	$66,453$	−1.73%	$36,318$	$35,188$	3.21%	85.11%	86.86%	−2.01%
Scenario 2×1	$48,940$	$47,960$	2.04%	$31,963$	$30,565$	4.57%	74.49%	75.04%	−0.73%
Scenario 2×0.95	$37,458$	$37,699$	−0.64%	$26,123$	$25,977$	0.56%	64.59%	65.31%	−1.10%
Scenario 3×1.05	$84,354$	$83,820$	0.64%	$33,202$	$31,448$	5.58%	87.18%	87.94%	−0.86%
Scenario 3×1	$65,319$	$65,962$	−0.97%	$28,937$	$28,423$	1.81%	77.65%	78.55%	−1.15%
Scenario 3×0.95	$52,677$	$52,654$	0.04%	$23,989$	$23,284$	3.03%	68.77%	69.10%	−0.48%
Scenario 4×1.05	$55,488$	$55,512$	−0.04%	$32,933$	$33,523$	−1.76%	77.69%	77.90%	−0.27%
Scenario 4×1	$41,550$	$42,347$	−1.88%	$28,066$	$28,754$	−2.39%	66.24%	66.71%	−0.70%
Scenario 4×0.95	$31,093$	$30,569$	1.71%	$21,804$	$21,840$	−0.16%	54.91%	54.01%	1.67%

7.3 Optimization Results

In this case, the production planning is considered as a two-objective optimization problem which seeks to minimize the expected total cost $\mathrm{E}[TC(\mathbf{x})]$ and maximize the expected demand fulfill rate $\mathrm{E}[DF(\mathbf{x})]$ simultaneously. As a multi-objective optimization problem with conflicting objectives, the goal is to find a set of non-dominated solutions that are as close as possible to the Pareto-optimal front and that are as diverse as possible [26] to allow decision makers to weigh the trade-offs.

As mentioned earlier, the MOGA function "gamultiobj" provided by Matlab is adopted to perform a search in the decision space of $\mathbf{x}$. The "gamultiobj" function employs the Elitism Non-Dominated Sorting GA (NSGA-II) algorithm, which is widely used for multi-objective optimization problems [26]. The user-specified parameters for the algorithm include: population size, maximum number of generations, stopping criteria, mutation function, elite count, and initial population. In our work, the population size and maximum number of generations are set as 75 and 8, respectively. The initial population has the same size as the population size, and is generated by combining a fractional factorial design and the default space-filling design in "gamultiobj" to provide a good coverage of the $\mathbf{x}$ space. To be specific, a 2^{8-2}_{V} fractional factorial design is first conducted to generate 64 design points in the $\mathbf{x}$ space, the additional 11 design points are determined using the Matlab default design. The higher and lower levels used in the fractional factorial design correspond to system utilization 95% and 55% respectively. All other parameters are left as their default values in Matlab.

The production plan optimization was performed for the four demand scenarios (Figure 7.1) respectively. For each demand scenario, a number of non-dominated solutions were obtained, and each solution's performance pair ($\widehat{E}[TC(\mathbf{x})]$, $\widehat{E}[DF(\mathbf{x})]$) is plotted in Figure 7.3. The obtained four Pareto fronts of non-dominated solutions provide similar coverage of the performance region: The expected total cost ranges from $12,879$ to $194,925$, while the demand fulfill rate spans from 18.82% to 99.91%. Such Pareto fronts are able to provide



(a) Scenarios 1 and 2

(b) Scenarios 3 and 4

Figure 7.3: Performance metrics of the non-dominated solutions obtained from the multi-objective optimization for each demand scenarios: "∘" corresponds to Scenarios 1 or 3; "∗" corresponds to Scenarios 2 or 4.

decision makers relatively complete information regarding the trade-offs between cost and customer satisfaction.

As shown in Figure 7.3 (a) or (b), the Pareto front depicted by the stars (for Scenarios 2 or 4) is slightly higher than that represented by the circles (for Scenarios 1 or 3): With the same cost, a higher demand fulfill rate can be achieved for Scenarios 2 (or 4), compared to Scenarios 1 (or 3). This is mainly because that the plan optimization is able to match the job outputs with the more frequently-realized (biweekly) customer demand in Scenarios 2 (or 4), leading to lower finished-goods inventory than the cases (Scenarios 1 or 3) where demand is realized weekly.

Table 7.2 provides for each demand scenario, two non-dominated solutions (represented by eight-element row vectors in the table) whose demand fulfill rates are respectively around 95% and 99%, the high levels that are usually of interest. For comparison purpose, the release plan that keeps the system running at 95% utilization over the planning horizon is also given in Table 7.2, with the expected number of jobs released each week being $(120, 120, 120, 120, 120, 120, 120, 120)$ during the period; this release plan leads to very high demand fulfill rate and very high cost, as expected. The selected solutions (that is, the row vectors in Table 7.2 representing the expected number of weekly-released jobs over the eight-week horizon) are compared with the expected weekly demand under their corresponding demand scenarios. It appears that to achieve a demand fulfill rate of about 95%, these solutions tend to follow the same general trend of customer demand (increasing for Scenarios 1 and 2, and fluctuating for Scenarios 3 and 4), but almost always over-produce a little in each week. This pattern becomes less obvious in the solutions leading to a demand fulfill rate of about 99%, which is achieved by more pronounced over-production at substantially higher total cost.

Table 7.2: Selected non-dominated solutions obtained from multi-objective optimization for each demand scenario.

	Expected weekly demand of Scenarios 1 and 2									
	69	76	82	88	95	101	107	113		
	Expected number of jobs released each week								$\widehat{E}[TC(\mathbf{x})]$	$\widehat{\mathrm{E}}[DF(\mathbf{x})]$
Scenario 1	76	79	84	92	108	103	109	117	97,697	93.88%
	101	90	78	118	100	114	112	105	153,978	99.15%
	120	120	120	120	120	120	120	120	259,128	100.00%
Scenario 2	88	81	85	96	110	104	111	106	90,548	93.96%
	90	86	106	119	120	113	116	116	141,450	99.10%
	120	120	120	120	120	120	120	120	229,177	100.00%
	Expected weekly demand of Scenarios 3 and 4									
	76	95	113	139	107	88	69	88		
	Average number of jobs released each week								$\widehat{E}[TC(\mathbf{x})]$	$\widehat{\mathrm{E}}[DF(\mathbf{x})]$
Scenario 3	88	118	115	119	109	110	76	83	112,235	94.85%
	116	117	119	120	112	117	96	119	159,374	99.24%
	120	120	120	120	120	120	120	120	178,017	99.51%
Scenario 4	101	118	114	114	111	114	86	96	98,539	94.33%
	117	120	120	115	115	120	99	102	132,431	98.24%
	120	120	120	120	120	120	120	120	146,337	98.77%

Chapter 8

Summary and Discussion

This dissertation developed a metamodel-based Monte Carlo simulation (MCS) method to address a fundamental issue for optimization of production planning: the quantification of the relationships between a release plan of jobs and its resulting performance metrics (e.g., the total cost involved and customer demand fulfill rate), which serves as the basis to find an optimum or good release plan leading to a best system performance. Such relationships result from an input release plan triggering the manufacturing system's non-stationary time-series outputs (WIP and job departures), and subsequently from the output processes interacting with the customer demand, another general time series. The existing approaches are inadequate to capture such relationships for responsive decision making: Analytical methods lack the high fidelity to real systems, while discrete-event simulation (DES) is typically too computationally intensive to run in real time. The MCS method is able to overcome the lack of fidelity of analytical methods and the computational burden of DES, allows for an accurate and quick evaluation of a release plan in terms of its performance metrics, which include not only the expectations (e.g., the mean cost) but also the variances (e.g., the variance of the cost) and probabilities of interest (e.g., the demand fulfill rate). The evaluation time requested by the MCS is independent of the complexity of the real systems being investigated.

The real-time "what if" analysis rendered by the MCS method provides the necessary basis for the optimization of production planning in a timely manner. The MCS-based multi-objective optimization problem was solved for the production planning of a scaled-down semiconductor fabrication system. The evaluation quality of the MCS is demonstrated by comparing the MCS estimates with those given by DES. The non-dominated solutions obtained from the multi-objective optimization allow decision makers to weigh the trade-offs between conflicting objectives such as minimizing the expected total cost and maximizing the demand fulfill rate.

Currently, the material requirement planning (MRP) is the most widely used production planning systems used in manufacturing industry. It can be embedded into an optimization loop to find optimal plan that minimizes the total cost. MRP is usually flawed by ignoring the uncertainty in the manufacturing system and using fixed job lead time disregarding the system congestion information. Comparing to MRP, the proposed approach is able to fully capture the stochastic behaviour of the system and integrate with a MOGA algorithm to find good release plans that consider the cost and customer service performance simultaneously. Before the proposed approach can be used in manufacturing industry, there are mainly three issues need to be solved in future research.

First, The proposed approach only deals with single-product system. In order to handle multi-product system, the metamodel must be extended to handle more complex system input-output relationships. To be specific, the metamodel inputs should include the release rates of each type of products (jobs), and the metamodel outputs should include the characteristics of the WIP and departure processes of each type of products, as well as the interactions between different types of products. Thus the number of regression coefficients in this extended metamodel will be huge and there may exist potential problem lies in the model selection and fitting process.

Second, the proposed approach doesn't consider the component parts needed for each unit of product. An potential improvement on the current approach is to take into account the bill of materials (BOM), and integrate the inventory, purchase and cost of component parts of each type of products into the production planning framework.

Third, once there is a change on the manufacturing system, a new metamodel need to be fit by collecting extensive offline DES data, which will keep the DES model running for weeks or even months. However, if there are frequently minor changes on the system, such as adding a machine in a workstation or adjusting the mean time to repair for an unreliable machine, it is hard to make the decision on whether to fit a new metamodel to accommodate these changes.

Appendix A

Configuration of the Example System

The example system considered in the empirical study is a scaled-down semiconductor fabrication system developed by Kayton et al. [58]. It consists of 9 workstations, and includes the major features of real semiconductor fabrication system [58] such as re-entrant flows, machine failures, and batch processing. One type of jobs is considered in this work with job processing sequence shown in Figure A.1. Each job has 14 processing steps, and needs to visit workstation 1, 4 and 6 multiple times. The inter-arrival times of jobs are assumed to follow exponential distribution with arrival rate specified by the job release plan. Table A.1 gives the specific configuration of each of the 9 workstations. The first three rows of the table specify for each station the number of machines, batch processing size, and whether the machines are subject to random failures. Lognormal distribution is assumed for all the processing times (PT), and Weibull is adopted for all the times to failure (TTF) and times to repair (TTR). For each station, the means and standard deviations (Stdev) of PT, TTF and TTR are also provided in Table A.1.

Table A.1: Configuration of workstations.

Station Index	1	2	3	4	5	6	7	8	9
Number of Machines	1	1	1	2	1	1	1	1	1
Batch Size (Min/Max)	2/4	2/4	1	1	1	1	1	1	1
Failure	No	No	Yes	No	Yes	Yes	Yes	Yes	No
Mean of PT (minutes)	80	220	80	40	25	22	40	50	50
Stdev of PT(minutes)	7	16	7	4	2	2.4	4	4	5
Mean of TTF (minutes)	–	–	720	–	1100	1170	720	1333	–
Stdev of TTF (minutes)	–	–	720	–	1100	1170	720	1333	–
Mean of TTR (minutes)	–	–	108.3	–	117.4	126.4	108.3	180.5	–
Stdev of TTR (minutes)	–	–	73.6	–	79.7	85.8	73.6	122.6	–

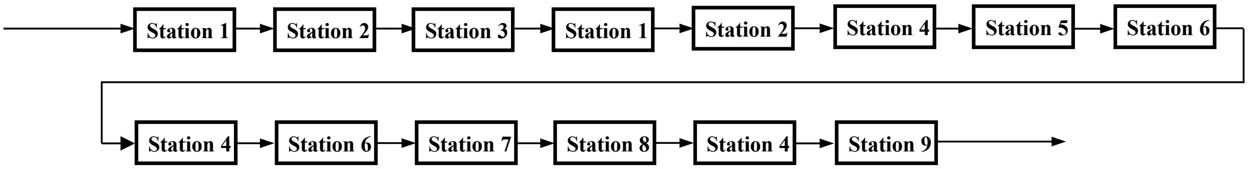


Figure A.1: Job processing sequence.

Appendix B

Design of Experiments for Estimating Metamodel

Assume that a discrete-event simulation (DES) model is available for the investigated system. In this part we discuss how to collect sample data via simulation experiments to fit the metamodel. For a simulation period with a length of L time units (Δt), each simulation replication is obtained by running the DES model under an arrival rate function $x(t)$. Multiple replications, say R replications, of the simulation runs need to be collected for the sample data. The arrival rate function $x(t)$, the simulation length L and the number of replications R are determined following the design of experiments (DOE) procedures in Yang and Liu [90] and given bellow.

Since simulation data during both steady state and transient period of the system need to be collected for the model fitting, a piecewise constant form with K distinct constant levels $\{x_1,x_2,...x_K\}$ is adopted for the arrival rate function $x(t)$. Before we specify the levels of $x(t)$, the concepts of capacity and utilization are introduced as follows. The system capacity η is defined as the maximum arrival rate the system can handle with long term stability; the system utilization under the arrival rate x_m equals to x_m/η which describes the fraction of busy time in the long run. Usually, the utilization is in the range of $(0, 1)$ and the system behaviour under high utilization receives more concern than that under the low utilization in the real world. The utilization range of interest is denoted as $[\rho_L, \rho_H]$ which is specified by users and the corresponding range of the $x(t)$ is denoted as:

$$[x_L, x_H] = [\rho_L\eta, \rho_H\eta] \tag{B.1}$$

As mentioned by Yang et al. [90], $K = 5$ is recommended for $x(t)$ and the steady state behaviour of the system can be captured by spreading the 5 distinct levels evenly across the range of $[x_L, x_H]$. In order to examine the interaction effects between $x(t)$ and $\mathbf{y}(t)$, it is also recommended that the 5 arrival rates are sequenced such that the difference between two successive levels of arrival rates is maximized.

Considering that the metamodel is required to capture both steady state and transient behaviour of the system, the total simulation length L should be determined in a manner such that the length of the steady state period is close to that of the transient period for the sample data. Let l_{tr} be the length of the time that an initially empty system needs to reach the steady state under the arrival rate x_L+$(x_L + x_H)/2$, which can be estimated through the simulation experiments following the methods in Law and Kelton [62]. In light of the discussion above, the period length during each level of $x(t)$ is determined to be $2l_{tr}$ and the total simulation length L for each replication would be given as $10l_{tr}$.

As the arrival rate function $x(t)$ and the simulation length L have been specified, we follow the two-step procedure used in [90] to determine the number of replications R as follows. First an initial number of replications R_0 are generated and the estimated $m_1(t)$ are denoted as $\widehat{m}_1^{(0)}(t)$. Let

$$\widehat{\sigma}_{\max}^{(0)} = \max_{t=1,2,\ldots,L} \widehat{\sigma}(\widehat{m}_1^{(0)}(t))$$

be the maximum sample standard deviation of $\widehat{m}_1^{(0)}(t)$ over $(0, L]$ resulted from the R_0 replications. Then the required number of replications R can be estimated as:

$$R = \lceil (\widehat{\sigma}_{\max}^{(0)})^2 / (\widehat{m}_1^{(0)}(t_{\max}) \times \gamma\%)^2 \rceil$$

where $t_{\max}$ is the time that achieves $\widehat{\sigma}_{\max}^{(0)}$ and $\gamma\%$ is the desired precision. Hence at the second step, $R - R_0$ additional replications are carried out such that the desired precision $\gamma\%$ can be reached.

Appendix C

Model Fitting and Selection of Metamodel

Based on the DES sample data, the model (4.1) is estimated and rewritten as

$$\widehat{\mathbf{y}}(t) = \widehat{\mathbf{F}}(\boldsymbol{\theta}; \widehat{x}(t), \widehat{x}(t-1), ..., \widehat{\mathbf{y}}(t-1), \widehat{\mathbf{y}}(t-2), ...) + \mathbf{e}(t) \tag{C.1}$$

where $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_{J_Q+J_D+5}\}$ and $\mathbf{e}(t) = \{e_1(t), e_2(t), ..., e_{J_Q+J_D+5}(t)\}$ denote the parameters to be estimated and the disturbance for function $\{F_i;\ i = 1, 2, ..., J_Q + J_D + 5\}$ respectively. The $\mathbf{e}(t)$ are added into the model to take into account the random errors involved in the estimation of the $\widehat{x}(t)$ and $\widehat{\mathbf{y}}(t)$. Following the transient queuing analysis in [90], we assume that each F_i takes the form of polynomial containing the main and interaction effects of the historical input and outputs. To be specific, each F_i in (C.1) takes the form of:

$$y_i(t) = \sum_{j_1=0}^{V_1} \sum_{j_2=0}^{V_1-j_1} \cdots \sum_{j_{V_2}=0}^{V_1-\sum_{l=1}^{V_2-1} j_l} \sum_{h=1}^{V_1-\sum_{l=1}^{V_2} j_l} b_{ij_1j_2\cdots j_{V_2}h} y_1^{j_1}(t-1) y_2^{j_2}(t-1) \cdots y_{V_2}^{j_{V_2}}(t-1) x^h(t-1) \tag{C.2}$$

for $i = 1, 2, ..., V_2$, where V_1 is the highest order of the polynomial terms and V_2 is the dimension of $\mathbf{y}(t)$. In this work, V_1 is set to be 2 based on the authors' empirical experience and V_2 equals to $J_Q + J_D + 5$. The model (C.2) at this point is the most complicated one and usually contains insignificant terms, thus the backward stepwise elimination procedure adopted by [90] will be used for each F_i to obtain a reduced model with a better fit. At each step of the elimination, the candidate model are fitted to the sample data $\{\widehat{x}(t), \widehat{\mathbf{y}}(t),\ t = 1, 2, ..., L\}$ by the least square methods [65].

Appendix D

Estimation and Evaluation of Metamodel

$$\widehat{m}_1(t) = m_1(t-1) - d_1(t-1) + x(t-1) \tag{D.1}$$

$$\begin{aligned}\widehat{d_1}(t) = {} & 0.0179 + 0.0434 m_1(t-1) + 0.4561 d_1(t-1) - 0.0066 m_2(t-1) \\ & + 0.0071 m_{12}(t-1) - 0.1449 d_{11}(t-1) - 0.0093 m_1(t-1) d_2(t-1) \\ & - 0.0034 m_1(t-1) d_{11}(t-1) + 0.4666 d_1(t-1)^2 + 0.0072 d_1(t-1) m_2(t-1) \\ & - 0.0082 d_1(t-1) m_{11}(t-1) + 0.0004 m_2(t-1) d_{11}(t-1) + 0.0741 d_2(t-1) x(t-1) \\ & - 0.0909 d_{12}(t-1) x(t-1) - 0.0307 x(t-1)^2\end{aligned} \tag{D.2}$$

$$\begin{aligned}\widehat{m}_2(t) = {} & 0.5662 - 0.1149 m_1(t-1) - 0.7456 d_1(t-1) + 1.0536 m_2(t-1) \\ & - 0.1939 m_{11}(t-1) + 0.1469 m_{12}(t-1) - 2.0810 e_{QD}(t-1) + 2.7935 x(t-1) \\ & + 0.0119 m_1(t-1)^2 + 1.8148 m_1(t-1) x(t-1) + 1.0695 d_1(t-1) d_2(t-1) \\ & - 1.2257 d_1(t-1) x(t-1) + 0.0024 m_2(t-1) e_{QD}(t-1) - 0.0026 m_{12}(t-1) e_{QD}(t-1) \\ & - 0.5545 d_{11}(t-1) x(t-1) + 1.3218 x(t-1)^2\end{aligned} \tag{D.3}$$

$$\begin{aligned}\widehat{d}_2(t) = {} & 0.0875 + 0.0278m_1(t-1) + 0.7497d_1(t-1) + 0.0304m_1d_1(t-1) \\ & - 0.0155m_1(t-1)d_1(t-1) - 0.0215m_1(t-1)d_{11}(t-1) + 0.3201d_1(t-1)^2 \\ & + 0.2443d_1(t-1)d_2(t-1) + 0.1496d_2(t-1)x(t-1) - 0.1550d_{12}(t-1)x(t-1) \\ & - 0.1007x(t-1)^2\end{aligned} \tag{D.4}$$

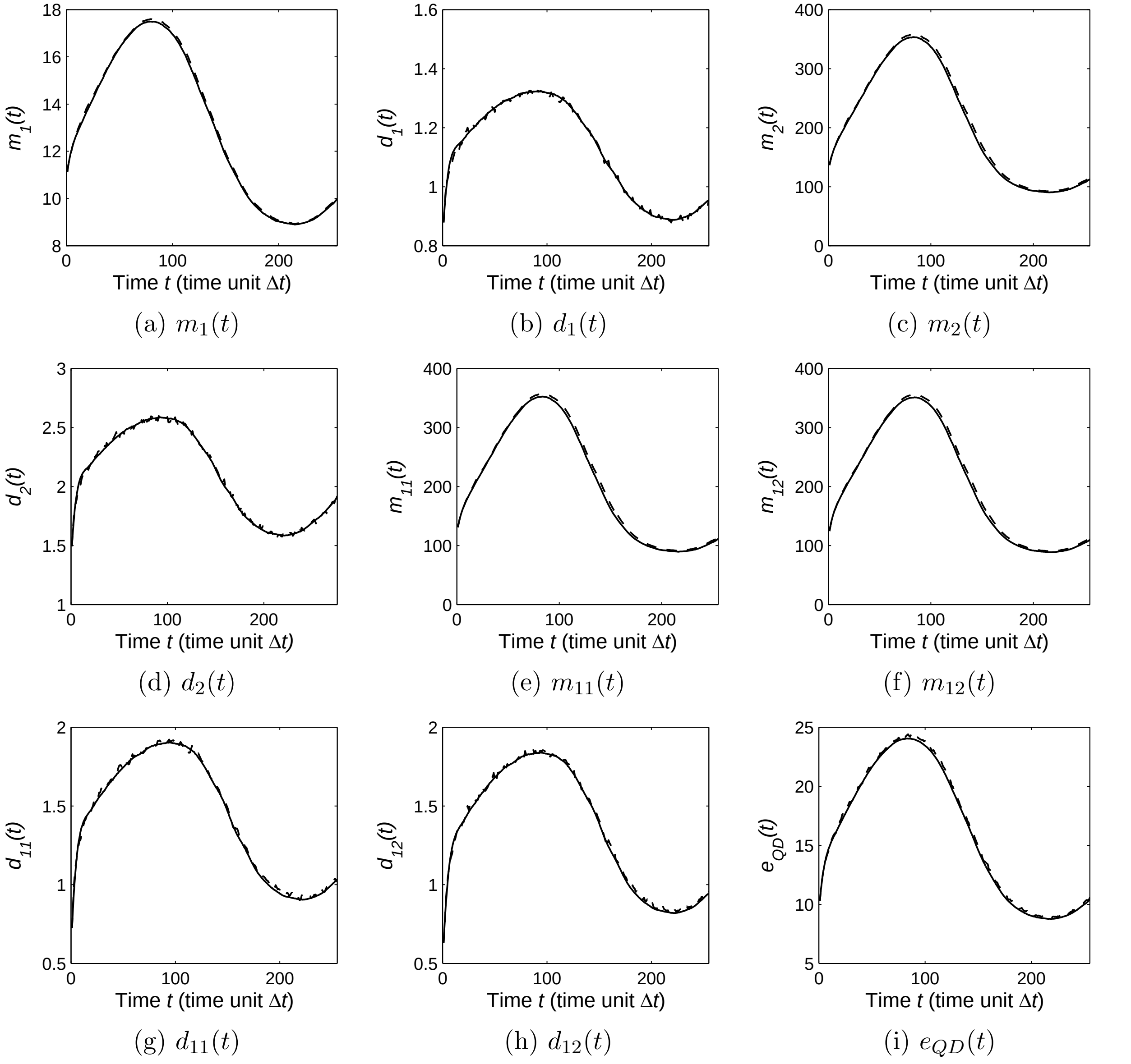
$$\begin{aligned}\widehat{m}_{11}(t) = {} & 0.1792 - 0.0852m_1(t-1) + 0.2214d_1(t-1) + 1.0823m_2(t-1) \\ & - 0.1283m_{11}(t-1) + 0.0465m_{12}(t-1) - 0.9557e_{QD}(t-1) + 0.0044m_1(t-1)^2 \\ & 0.9168m_1(t-1)x(t-1) - 0.00003m_{12}(t-1)e_{QD}(t-1) + 0.3236x(t-1)^2\end{aligned} \tag{D.5}$$

$$\begin{aligned}\widehat{m}_{12}(t) = {} & 0.4224 - 0.0820m_1(t-1) - 0.0424d_1(t-1) + 0.1221m_2(t-1) \\ & - 0.1556d_2(t-1) + 0.8277m_{11}(t-1) + 0.0438m_{12}(t-1) - 0.3298d_{11}(t-1) \\ & - 0.8196e_{QD}(t-1) - 0.3403x(t-1) + 0.0024m_1(t-1)d_{12}(t-1) \\ & + 1.0271m_1(t-1)x(t-1) - 0.3062d_1(t-1)d_{12}(t-1) + 0.0124d_1(t-1)e_{QD}(t-1) \\ & - 0.0018m_2(t-1)^2 + 0.0035m_2(t-1)m_{11}(t-1) + 0.2436d_2(t-1)x(t-1) \\ & - 0.0017m_{11}(t-1)^2 - 0.0001m_{12}(t-1)e_{QD}(t-1) - 0.5429x(t-1)^2\end{aligned} \tag{D.6}$$

$$
\begin{aligned}
\widehat{d}_{11}(t) = & -0.0916 - 0.0294 m_1(t-1) + 0.4253 d_1(t-1) - 0.0015 m_2(t-1) \\
& + 0.0886 e_{QD}(t-1) + 0.0330 m_1(t-1) d_1(t-1) - 0.0101 m_1(t-1) d_2(t-1) \\
& + 0.1408 d_1(t-1) x(t-1) + 0.0080 m_2(t-1) x(t-1) + 0.1928 d_2(t-1) d_{11}(t-1) \\
& - 0.0153 m_{11}(t-1) x(t-1) + 0.0083 m_{12}(t-1) x(t-1) - 0.0173 d_{12}(t-1) e_{QD}(t-1) \\
& - 0.0261 e_{QD}(t-1) x(t-1) + x(t-1)^2
\end{aligned}
\tag{D.7}
$$

$$
\begin{aligned}
\widehat{d}_{12}(t) = & -0.1667 + 0.0103 m_1(t-1) + 0.2946 d_1(t-1) - 0.0116 m_2(t-1) \\
& + 0.0105 m_{12}(t-1) + 0.0699 e_{QD}(t-1) - 0.0050 m_1(t-1) d_2(t-1) \\
& + 0.0081 m_1(t-1) x(t-1) + 0.0051 d_1(t-1) m_2(t-1) - 0.0062 d_1(t-1) m_{12}(t-1) \\
& + 0.4627 d_1(t-1) d_{12}(t-1) + 0.0011 m_{12}(t-1) d_{12}(t-1) - 1.4809 d_{11}(t-1)^2 \\
& + 3.3272 d_{11}(t-1) d_{12}(t-1) - 0.0605 d_{11}(t-1) x(t-1) - 1.7835 d_{12}(t-1)^2 \\
& - 0.0285 d_{12}(t-1) e_{QD}(t-1)
\end{aligned}
\tag{D.8}
$$

$$
\begin{aligned}
\widehat{e}_{QD}(t) = & -0.6245 + 0.1760 m_1(t-1) + 0.0134 m_{12}(t-1) - 2.4162 d_{12}(t-1) \\
& - 0.2578 m_1(t-1) d_1(t-1) + 0.8689 m_1(t-1) d_{11}(t-1) - 0.4930 m_1(t-1) d_{12}(t-1) \\
& - 0.1511 m_1(t-1) x(t-1) + 6.2565 d_1(t-1) x(t-1) - 0.0275 m_{11}(t-1) d_{11}(t-1) \\
& + 0.0217 m_{12}(t-1) d_{11}(t-1) - 9.5731 d_{11}(t-1) x(t-1) + 0.7797 e_{QD}(t-1) \\
& - 0.1312 d_{12}(t-1) e_{QD}(t-1) + 6.1842 d_{12}(t-1) x(t-1) + 0.9478 d_{12}^2 \\
& - 0.6269 x(t-1)^2
\end{aligned}
\tag{D.9}
$$

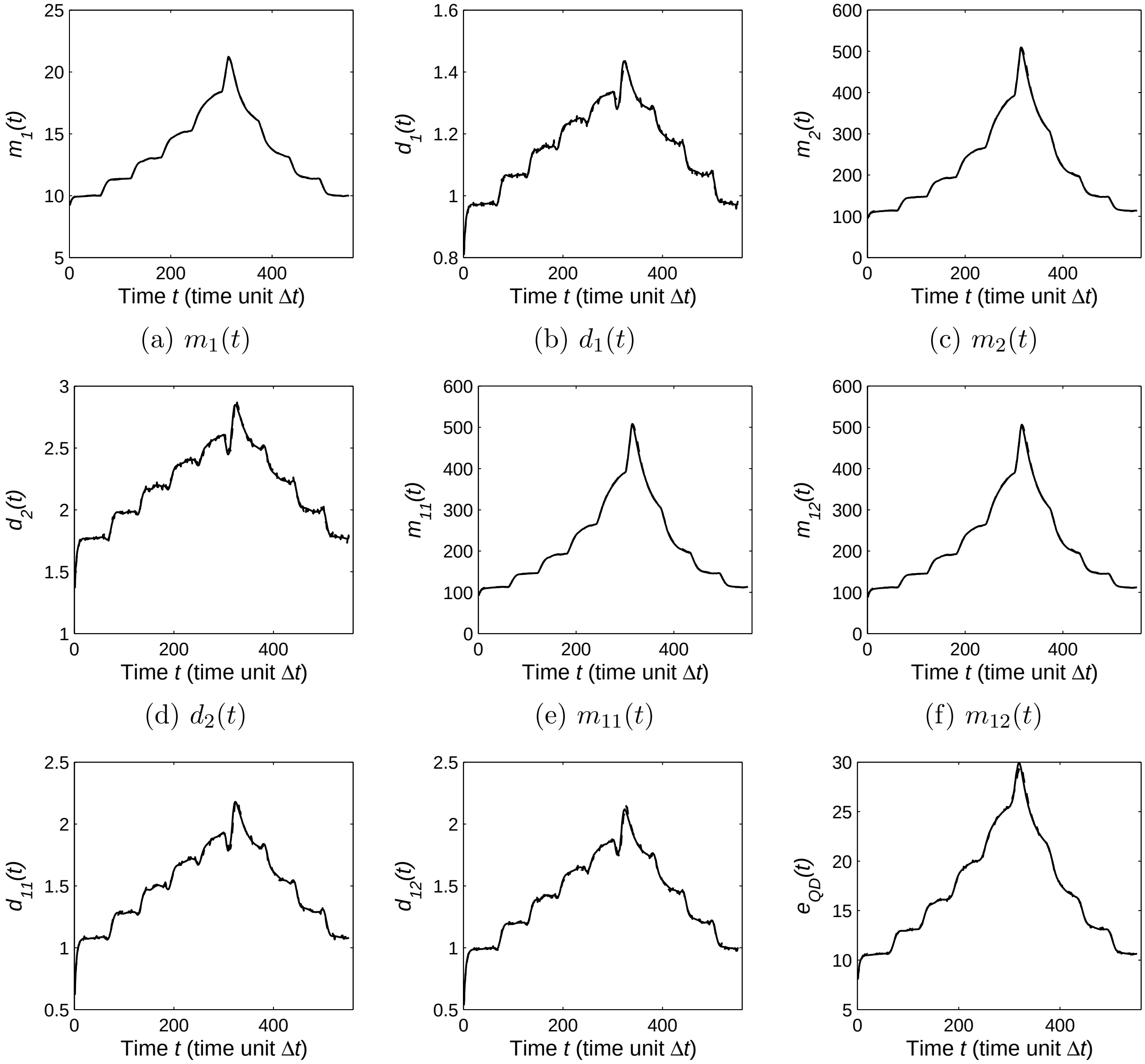


(a) $m_1(t)$ (b) $d_1(t)$ (c) $m_2(t)$

(d) $d_2(t)$ (e) $m_{11}(t)$ (f) $m_{12}(t)$

(g) $d_{11}(t)$ (h) $d_{12}(t)$ (i) $e_{QD}(t)$

Figure D.1: Evaluation of the fitted metamodel using VDS 1: dashed curves denote the "true" values and solid curves represent the metamodel-predicted results.



(a) $m_1(t)$ (b) $d_1(t)$ (c) $m_2(t)$

(d) $d_2(t)$ (e) $m_{11}(t)$ (f) $m_{12}(t)$

(g) $d_{11}(t)$ (h) $d_{12}(t)$ (i) $e_{QD}(t)$

Figure D.2: Evaluation of the fitted metamodel using VDS 2: dashed curves denote the "true" values and solid curves represent the metamodel-predicted results.

Appendix E

Yule-Walker Estimator of Autoregressive Processes

Let $Z(t)$ be an autoregressive process representing one of the two stationary processes from steady-state simulation: WIP $Q(t)$ and departures $D(t)$. The stationary process $Z(t)$ can be written in the autoregressive form

$$Z(t) = \sum_{j=1}^{J} \alpha_j Z(t-j) + \varepsilon(t); \qquad t = 1, 2, \ldots, N,$$

where J denotes the autoregressive order, $\{\alpha_j; j = 1, 2, \ldots, J\}$ the autoregressive coefficients, $\varepsilon(t)$ the innovation term, and N the number of observations.

The model selection in Section 5.1.2 between N-INAR and N-SINAR is made by performing the following hypothesis test

$$H_0 : \alpha_j \geq 0; \quad vs. \quad H_a : \alpha_j < 0 \tag{E.1}$$

for $j = 1, 2, \ldots, J$. If H_0 is rejected for any j, then N-SINAR will be selected; otherwise, N-INAR will be adopted. Yule-Walker estimation and inference are performed in this work to test the hypothesis (E.1), based on the steady-state DES data $\{Z_r(t); t = 1, 2..., N; r = 1, 2, ..., R\}$, where R is the number of replications. Let

$$\widetilde{Z}(t) = \frac{1}{R} \sum_{r=1}^{R} Z_r(t).$$

By central limit theorem, $\widetilde{Z}(t)$ approximately follows normal distribution with

$$\begin{aligned} \mathrm{E}[\widetilde{Z}(t)] &= \mathrm{E}[Z(t)], & \mathrm{Var}[\widetilde{Z}(t)] &= \frac{1}{R}\mathrm{Var}[Z(t)], \\ \mathrm{Cov}[\widetilde{Z}(t+j), \widetilde{Z}(t)] &= \tfrac{1}{R}\mathrm{Cov}[Z(t+j), Z(t)], & \mathrm{Corr}[\widetilde{Z}(t+j), \widetilde{Z}(t)] &= \mathrm{Corr}[Z(t+j), Z(t)] \end{aligned}$$

when R is large. Hence, $\widetilde{Z}(t)$ can be modeled by a AR(J) process

$$\widetilde{Z}(t) = \sum_{j=1}^{J} \alpha_j \widetilde{Z}(t-j) + \varepsilon(t); \qquad t = 1, 2, \ldots, N$$

where $\varepsilon(t)$ is the i.i.d normally-distributed innovation term with $\mathrm{Var}[\varepsilon(t)] = \sigma^2$. Denote

$$\gamma_0 = \mathrm{Var}[\widetilde{Z}(t)], \;\; \mathrm{Cov}[\widetilde{Z}(t+j), \widetilde{Z}(t)] = \gamma_j,$$

the Yule-Walker equation [14, 16] is written as

$$\boldsymbol{\Gamma}\boldsymbol{\alpha} = \boldsymbol{\gamma}, \tag{E.2}$$

where

$$\boldsymbol{\Gamma} = \begin{pmatrix} \gamma_0 & \gamma_1 & \gamma_2 & \cdots & \gamma_{J-1} \\ \gamma_1 & \gamma_0 & \gamma_1 & \cdots & \gamma_{J-2} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \gamma_{J-1} & \gamma_{J-2} & \gamma_{J-3} & \cdots & \gamma_0 \end{pmatrix}, \;\; \boldsymbol{\alpha} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_J \end{pmatrix}, \;\; \boldsymbol{\gamma} = \begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_J \end{pmatrix}.$$

Replacing $\{\gamma_0, \gamma_1, ..., \gamma_J\}$ by the sample estimates $\{\widehat{\gamma}_0, \widehat{\gamma}_1, ..., \widehat{\gamma}_J\}$ in (E.2), $\boldsymbol{\alpha}$ and σ^2 can be estimated as

$$\widehat{\boldsymbol{\alpha}} = \widehat{\boldsymbol{\Gamma}}^{-1}\widehat{\boldsymbol{\gamma}}, \;\; \widehat{\sigma}^2 = \widehat{\gamma}_0 - \widehat{\boldsymbol{\alpha}}^\top \widehat{\boldsymbol{\gamma}}.$$

Since $Z(t)$ and $\widetilde{Z}(t)$ have the same autocorrelation structure, it can be shown that these two processes have the same Yule-Walker estimates for $\boldsymbol{\alpha}$ [88, 55]. The hypothesis (E.1) is performed based on $\widetilde{Z}(t)$.

When the sample size N is large, the Yule-Walker estimator $\widehat{\boldsymbol{\alpha}}$ of $\widetilde{Z}(t)$ follows asymptotic normal distribution [16, 17]

$$\widehat{\boldsymbol{\alpha}} \sim \mathcal{N}(\boldsymbol{\alpha}, N^{-1}\sigma^2\boldsymbol{\Gamma}^{-1}),$$

and

$$\frac{\widehat{\alpha}_j - \alpha_j}{\sqrt{\widehat{w}_{jj}/N}} \sim \mathcal{N}(0, 1)$$

where $\widehat{w}_{jj}$ denotes the j^{th} diagonal element of $\widehat{\sigma}^2\widehat{\boldsymbol{\Gamma}}^{-1}$. Denote Φ as the the cumulative distribution function of the standard normal distribution. For the hypothesis test (E.1) with significance level δ, if $\frac{\widehat{\alpha}_j}{\sqrt{\widehat{w}_{jj}/N}}$ is less than $\Phi^{-1}(\delta)$, H_0 will be rejected; otherwise, H_0 is considered acceptable.

Appendix F

Model Fitting of Bivariate Time Series

For the bivariate output process $(Q(t),\ D(t))^\top$ driven by an arrival process, the fitted metamodel is able to predict its characteristics

$$\widehat{\mathbf{y}}(t) = (\widehat{m}_1(t), \widehat{m}_2(t), \widehat{d}_1(t), \widehat{d}_2(t), \widehat{m}_{11}(t), \ldots, \widehat{m}_{1J_Q}(t), \widehat{d}_{11}(t), \ldots, \widehat{d}_{1J_D}(t), \widehat{e}_{QD}(t))^\top. \tag{F.1}$$

Based on $\widehat{\mathbf{y}}(t)$, how to fit a bivariate time-series model for $(Q(t),\ D(t))^\top$, with the identified time-series model family (Section 5.1.3)?

F.1 Properties of Autoregressive Models

The general form of the candidate autoregressive models (5.1) identified for output processes is re-written below for convenience

$$Z(t) = \sum_{j=1}^{J} \mathcal{R}(\alpha_j(t), Z(t-j)) + \varepsilon(t); \qquad t = 1, 2, \ldots. \tag{F.2}$$

As discussed in Section 5.1.1, the random operator $\mathcal{R}$ can be scalar multiplication, binomial thinning "$\circ$" or generalized thinning "$F(\cdot)\circ$", corresponding to N-AR, N-INAR and N-SINAR model respectively. A time series $Z(t)$ following any of the three models has the following properties [14][28][88][55]:

$$\mathrm{E}[\mathcal{R}(\alpha_j(t), Z(t-j))] = \alpha_j(t)\mathrm{E}[Z(t-j)] \tag{F.3}$$

$$\mathrm{E}[\mathcal{R}(\alpha_j(t), Z(t-j))^2] = (\alpha_j(t))^2\mathrm{E}[(Z(t-j))^2] + f_{\mathcal{R}}(\alpha_j(t), \mathrm{E}[Z(t-j)]) \tag{F.4}$$

$$\mathrm{Var}[\mathcal{R}(\alpha_j(t), Z(t-j))] = (\alpha_j(t))^2\mathrm{Var}[Z(t-j)] + f_{\mathcal{R}}(\alpha_j(t), \mathrm{E}(Z(t-j))) \tag{F.5}$$

$$\mathrm{E}[\mathcal{R}(\alpha_j(t), Z(t-j)), \mathcal{R}(\alpha_i(t), Z(t-i))] = \alpha_j(t)\alpha_i(t)\mathrm{E}[Z(t-j)Z(t-i)] \tag{F.6}$$

$$\mathrm{Cov}[\mathcal{R}(\alpha_j(t), Z(t-j)), \mathcal{R}(\alpha_i(t), Z(t-i))] = \alpha_j(t)\alpha_i(t)\mathrm{Cov}[Z(t-j), Z(t-i)]. \tag{F.7}$$

The form of $f_{\mathcal{R}}(\cdot)$ in (F.4) and (F.5) is model-dependent: For N-AR, $f_{\mathcal{R}}(\alpha_j(t), \mathrm{E}[Z(t-j)]) = 0$; for N-INAR, $f_{\mathcal{R}}(\alpha_j(t), \mathrm{E}[Z-j]) = \alpha_j(t)(1-\alpha_j(t))\mathrm{E}[Z(t-j)]$; and for N-SINAR, $f_{\mathcal{R}}(\alpha_j(t), \mathrm{E}[Z(t-j)]) = (1-\alpha_j(t)^2)\mathrm{E}[Z(t-j)]$.

Based on (F.3)–(F.7), some properties of the time series $Z(t)$ and the innovation process $\varepsilon(t)$ can be derived as follows. Specifically, we have

$$\begin{aligned} \mathrm{Cov}(Z(t), Z(t-j)) &= \sum_{i=1}^{J} \mathrm{Cov}[\mathcal{R}(\alpha_i(t), Z(t-i)), Z(t-j)] + \mathrm{Cov}[\varepsilon(t), Z(t-j)] && \text{(F.8)} \\ &= \sum_{i=1}^{J} \alpha_i(t)\mathrm{Cov}[Z(t-i), Z(t-j)]. && \text{(F.9)} \end{aligned}$$

Step (F.8) is obtained by respectively expressing $Z(t)$ and $Z(t-j)$ in terms of (F.2). Step (F.9) employs the property (F.7) and the assumption that $Z(t-j)$ is independent of $\varepsilon(t)$.

By taking expectation and variance on both sides of (F.2) and using the properties (F.3) and (F.5), the mean and variance of $\varepsilon(t)$ can be expressed in terms of the characteristics of $Z(t)$ as follows:

$$
\begin{aligned}
\mathrm{E}[\varepsilon(t)] &= \mathrm{E}[Z(t)] - \sum_{i=1}^{J} \alpha_i(t)\mathrm{E}[Z(t-i)] && \text{(F.10)} \\
\mathrm{Var}[\varepsilon(t)] &= \mathrm{Var}[Z(t)] - \sum_{i=1}^{J} \alpha_i^2(t)\mathrm{Var}[Z(t-i)] - \sum_{i=1}^{J} f_{\mathcal{R}}(\alpha_i(t), \mathrm{E}[Z(t-i)]) \\
&\quad -2\sum_{i=1}^{J-1}\sum_{j=i+1}^{J} \alpha_i(t)\alpha_j(t)\mathrm{Cov}[Z(t-i), Z(t-j)]. && \text{(F.11)}
\end{aligned}
$$

As will be seen below, the properties (F.9)–(F.11) enable the derivation from the metamodel-predicted characteristics (F.1) to the fitted parameters that specify the bivariate time-series model.

F.2 Fitting the Bivariate Time-Series Model

The following expectations, variances and covariances can be easily obtained from the metamodel prediction (F.1):

$$
\begin{aligned}
\widehat{\mathrm{E}}[\widehat{Q}(t)] &= \widehat{m}_1(t) && \text{(F.12)} \\
\widehat{\mathrm{Var}}[\widehat{Q}(t)] &= \widehat{m}_2(t) - (\widehat{m}_1(t))^2 && \text{(F.13)} \\
\widehat{\mathrm{Cov}}[\widehat{Q}(t), \widehat{Q}(t-j)] &= \widehat{m}_{1j}(t) - \widehat{m}_1(t)\widehat{m}_1(t-j) \text{ for } j = 1, 2, ..., J_Q && \text{(F.14)} \\
\widehat{\mathrm{E}}[\widehat{D}(t)] &= \widehat{d}_1(t) && \text{(F.15)} \\
\widehat{\mathrm{Var}}[\widehat{D}(t)] &= \widehat{d}_2(t) - (\widehat{d}_1(t))^2 && \text{(F.16)} \\
\widehat{\mathrm{Cov}}[\widehat{D}(t), \widehat{D}(t-j)] &= \widehat{d}_{1j}(t) - \widehat{d}_1(t)\widehat{d}_1(t-j) \text{ for } j = 1, 2, ..., J_D && \text{(F.17)} \\
\widehat{\mathrm{Cov}}[\widehat{Q}(t), \widehat{D}(t)] &= \widehat{e}_{QD}(t) - \widehat{m}_1(t)\widehat{d}_1(t) && \text{(F.18)}
\end{aligned}
$$

with $t = 1, 2, \ldots, H$.

Following the procedures in [14, 88], we use (F.12)–(F.18) to derive as follows the fitted model parameters for the bivariate time series: $\{\widehat{\alpha}_j^{(D)}(t); j = 1, 2, \ldots, J_D\}$, $\widehat{\mathrm{E}}[\widehat{\varepsilon}^{(Q)}(t)]$, $\widehat{\mathrm{E}}[\widehat{\varepsilon}^{(D)}(t)]$, $\widehat{\mathrm{Var}}[\widehat{\varepsilon}^{(Q)}(t)]$, $\widehat{\mathrm{Var}}[\widehat{\varepsilon}^{(D)}(t)]$, and $\widehat{\mathrm{Cov}}[\widehat{\varepsilon}^{(Q)}(t), \widehat{\varepsilon}^{(D)}(t)]$.

By employing (F.9), $\{\widehat{\alpha}_j^{(Q)}(t); j = 1, 2, \ldots, J_Q\}$, $\widehat{\mathrm{E}}[\widehat{\varepsilon}^{(Q)}(t)]$ and $\widehat{\mathrm{Var}}[\widehat{\varepsilon}^{(Q)}(t)]$ are derived as follows:

$$\widehat{\boldsymbol{\alpha}}^{(Q)}(t) = \begin{pmatrix} \widehat{\alpha}_1^{(Q)}(t) \\ \widehat{\alpha}_2^{(Q)}(t) \\ \vdots \\ \widehat{\alpha}_{J_Q}^{(Q)}(t) \end{pmatrix} = (\widehat{\mathbf{A}}^{(Q)}(t))^{-1}\widehat{\mathbf{b}}^{(Q)}(t), \tag{F.19}$$

where $\widehat{\mathbf{A}}^{(Q)}(t)$ is a $J_Q \times J_Q$ matrix with the element in the i^{th} row and j^{th} column being:

$$\widehat{\mathbf{A}}^{(Q)}(t)_{ij} = \widehat{\mathrm{Cov}}[\widehat{Q}(t - \min(i,j)), \widehat{Q}(t - \min(i,j) - |i-j|)], \quad 0 < i, j \leq J_Q,$$

and

$$\widehat{\mathbf{b}}^{(Q)}(t) = \begin{pmatrix} \widehat{\mathrm{Cov}}[\widehat{Q}(t), \widehat{Q}(t-1)] \\ \widehat{\mathrm{Cov}}[\widehat{Q}(t), \widehat{Q}(t-2)] \\ \vdots \\ \widehat{\mathrm{Cov}}[\widehat{Q}(t), \widehat{Q}(t-J_Q)] \end{pmatrix}.$$

Then, based on (F.10)–(F.11), we have

$$\widehat{\mathrm{E}}[\widehat{\varepsilon}^{(Q)}(t)] = \widehat{\mathrm{E}}[\widehat{Q}(t)] - \sum_{i=1}^{J_Q} \widehat{\alpha}_i^{(Q)}(t)\widehat{\mathrm{E}}[\widehat{Q}(t-i)] \tag{F.20}$$

$$\begin{aligned}\widehat{\mathrm{Var}}[\widehat{\varepsilon}^{(Q)}(t)] = {} & \widehat{\mathrm{Var}}[\widehat{Q}(t)] - \sum_{i=1}^{J_Q} (\widehat{\alpha}_i^{(Q)}(t))^2 \widehat{\mathrm{Var}}[\widehat{Q}(t-i)] \\ & - \sum_{i=1}^{J_Q} f_{\mathcal{R}^{(Q)}}(\widehat{\alpha}_i^{(Q)}(t), \widehat{\mathrm{E}}[\widehat{Q}(t-i)]) \\ & - 2\sum_{i=1}^{J_Q}\sum_{j>i}^{J_Q} \widehat{\alpha}_i^{(Q)}(t)\widehat{\alpha}_j^{(Q)}(t)\widehat{\mathrm{Cov}}[\widehat{Q}(t-i), \widehat{Q}(t-j)]\end{aligned} \tag{F.21}$$

The remaining parameter to be estimated is $\widehat{\mathrm{Cov}}[\widehat{\varepsilon}^{(Q)}(t),\, \widehat{\varepsilon}^{(D)}(t)]$, which leads to the specified $\widehat{\mathrm{Cov}}[\widehat{Q}(t), \widehat{D}(t)]$ given by (F.18). Assume $J_Q \geq J_D$ without loss of generality. The fitted bivariate model can be written as

$$\begin{pmatrix} \widehat{Q}(t) \\ \widehat{D}(t) \end{pmatrix} = \sum_{i=1}^{J_Q} \begin{pmatrix} \mathcal{R}^{(Q)}(\widehat{\alpha}_i^{(Q)}(t), \widehat{Q}(t-i)) \\ \mathcal{R}^{(D)}(\widehat{\alpha}_i^{(D)}(t), \widehat{D}(t-i)) \end{pmatrix} + \begin{pmatrix} \widehat{\varepsilon}^{(Q)}(t) \\ \widehat{\varepsilon}^{(D)}(t) \end{pmatrix} \tag{F.22}$$

where $\widehat{\alpha}_i^{(D)}(t) = 0$ for $i > J_D$. Denote

$$\widehat{\mathbf{\Sigma}}^{(QD)}(t) = \widehat{\mathrm{Var}}\left[\begin{pmatrix} \widehat{Q}(t) \\ \widehat{D}(t) \end{pmatrix}\right],\ \ \widehat{\mathbf{\Sigma}}^{(QD)}(t_1, t_2) = \widehat{\mathrm{Cov}}\left[\begin{pmatrix} \widehat{Q}(t_1) \\ \widehat{D}(t_1) \end{pmatrix}, \begin{pmatrix} \widehat{Q}(t_2) \\ \widehat{D}(t_2) \end{pmatrix}\right];\ \ t_1 \neq t_2,$$

$$\widehat{\mathbf{\Sigma}}^{(\widehat{\boldsymbol{\varepsilon}})}(t) = \widehat{\mathrm{Var}}\left[\begin{pmatrix} \widehat{\varepsilon}^{(Q)}(t) \\ \widehat{\varepsilon}^{(D)}(t) \end{pmatrix}\right],\ \ \widehat{\mathbf{\Lambda}}_i(t) = \begin{pmatrix} \widehat{\alpha}_i^{(Q)}(t) & 0 \\ 0 & \widehat{\alpha}_i^{(D)}(t) \end{pmatrix}.$$

Based on the properties (F.4)–(F.5), it can be shown that

$$\widehat{\mathrm{Var}}\left[\begin{pmatrix}\mathcal{R}^{(Q)}(\widehat{\alpha}_i^{(Q)}(t),\widehat{Q}(t-i))\\ \mathcal{R}^{(D)}(\widehat{\alpha}_i^{(D)}(t),\widehat{D}(t-i))\end{pmatrix}\right]=\widehat{\mathbf{\Lambda}}_i(t)\widehat{\mathbf{\Sigma}}^{(QD)}(t-i)\widehat{\mathbf{\Lambda}}_i(t)^\top + \begin{pmatrix}f_{\mathcal{R}^{(Q)}}(\widehat{\alpha}_i^{(Q)}(t),\widehat{E}[\widehat{Q}(t-i)]) & 0\\ 0 & f_{\mathcal{R}^{(D)}}(\widehat{\alpha}_i^{(D)}(t),\widehat{E}[\widehat{D}(t-i)])\end{pmatrix}. \tag{F.23}$$

Taking the variance on both sides of (F.22) and using (F.7), (F.23) and Lemma 2.1 in Du and Li [28], we have

$$\begin{aligned}\widehat{\mathbf{\Sigma}}^{(\widehat{\boldsymbol{\varepsilon}})}(t)=&-\sum_{i=1}^{J_Q}\widehat{\mathbf{\Lambda}}_i(t)\widehat{\mathbf{\Sigma}}^{(QD)}(t-i)\widehat{\mathbf{\Lambda}}_i(t)^\top-\sum_{i=1}^{J_Q}\sum_{j>i}^{J_Q}\widehat{\mathbf{\Lambda}}_i(t)\widehat{\mathbf{\Sigma}}^{(QD)}(t-i,t-j)\widehat{\mathbf{\Lambda}}_j(t)^\top\\&-\sum_{i=1}^{J_Q}\sum_{j>i}^{J_Q}\widehat{\mathbf{\Lambda}}_j(t)\widehat{\mathbf{\Sigma}}^{(QD)}(t-j,t-i)\widehat{\mathbf{\Lambda}}_i(t)^\top+\widehat{\mathbf{\Sigma}}^{(QD)}(t)\\&-\sum_{i=1}^{J_Q}\begin{pmatrix}f_{\mathcal{R}^{(Q)}}(\widehat{\alpha}_i^{(Q)}(t),\widehat{E}[\widehat{Q}(t-i)]) & 0\\ 0 & f_{\mathcal{R}^{(D)}}(\widehat{\alpha}_i^{(D)}(t),\widehat{E}[\widehat{D}(t-i)]).\end{pmatrix}\end{aligned}\tag{F.24}$$

Notice that $\widehat{\alpha}_i^{(D)}(t)=0$ for $i>J_D$, and $\widehat{\mathrm{Cov}}[\widehat{\varepsilon}^{(Q)}(t),\widehat{\varepsilon}^{(D)}(t)]$ is the off-diagonal element of $\widehat{\mathbf{\Sigma}}^{(\widehat{\boldsymbol{\varepsilon}})}(t)$ in (F.24) and can be written as

$$\begin{aligned}\widehat{\mathrm{Cov}}[\widehat{\varepsilon}^{(Q)}(t),\widehat{\varepsilon}^{(D)}(t)]=&\ \widehat{\mathrm{Cov}}[\widehat{Q}(t),\widehat{Z}(t)]-\sum_{i=1}^{J_D}\widehat{\alpha}_i^{(Q)}(t)\widehat{\mathrm{Cov}}[\widehat{Q}(t-i),\widehat{D}(t-i)]\widehat{\alpha}_i^{(D)}(t)\\&-\sum_{i=1}^{J_D}\sum_{j>i}^{J_D}\widehat{\alpha}_i^{(Q)}(t)\widehat{\mathrm{Cov}}[\widehat{Q}(t-i),\widehat{D}(t-j)]\widehat{\alpha}_j^{(D)}(t)\\&-\sum_{i=1}^{J_D}\sum_{j>i}^{J_D}\widehat{\alpha}_j^{(Q)}(t)\widehat{\mathrm{Cov}}[\widehat{Q}(t-j),\widehat{D}(t-i)]\widehat{\alpha}_i^{(D)}(t).\end{aligned}\tag{F.25}$$

In (F.25), $\widehat{\text{Cov}}[\widehat{Q}(t), \widehat{D}(t)]$ is a metamodel-predicted characteristics, and the remaining terms need to be estimated are

$$\widehat{\text{Cov}}[\widehat{Q}(t), \widehat{D}(t-i)], \;\; \widehat{\text{Cov}}[\widehat{Q}(t-i), \widehat{D}(t)], \;\; t = 1, 2, ..., H, \;\; i = 1, 2, ..., J_D. \tag{F.26}$$

Using the properties (F.3)-(F.7), and the facts that $\widehat{Q}(t-j)$ is independent of $\widehat{\varepsilon}^{(Q)}(t)$ in (5.11) and $\widehat{D}(t-j)$ is independent of $\widehat{\varepsilon}^{(D)}(t)$ in (5.12), the following results can be obtained:

$$\begin{aligned}\widehat{\text{Cov}}[\widehat{Q}(t), \widehat{D}(t-i)] &= \widehat{\text{Cov}}[\sum_{j=1}^{J_Q} \mathcal{R}^{(Q)}(\widehat{\alpha}_j^{(Q)}(t), \widehat{Q}(t-j)) + \widehat{\varepsilon}^{(Q)}(t), \widehat{D}(t-i)] \\ &= \sum_{j=1}^{J_Q} \widehat{\alpha}_j^{(Q)}(t) \widehat{\text{Cov}}[\widehat{Q}(t-j), \widehat{D}(t-i)]\end{aligned} \tag{F.27}$$

$$\begin{aligned}\widehat{\text{Cov}}[\widehat{Q}(t-i), \widehat{D}(t)] &= \widehat{\text{Cov}}[\widehat{Q}(t-i), \sum_{j=1}^{J_D} \mathcal{R}_j^{(D)}(\widehat{\alpha}_j^{(D)}(t), \widehat{D}(t-j)) + \widehat{\varepsilon}^{(D)}(t)] \\ &= \sum_{j=1}^{J_D} \widehat{\alpha}_j^{(D)}(t) \widehat{\text{Cov}}[\widehat{Q}(t-i), \widehat{D}(t-j)].\end{aligned} \tag{F.28}$$

The values of $\{\widehat{\text{Cov}}[\widehat{Q}(-i), \widehat{D}(-j)]; \; i, j = 0, 1, 2, ..., \max\{J_Q, J_D\} - 1\}$ can be typically obtained from the historical data to serve as the seed to initiate the computation in (F.27) and (F.28). And $\widehat{\text{Cov}}[\widehat{\varepsilon}^{(Q)}(t), \widehat{\varepsilon}^{(D)}(t)]$ can be obtained by plugging (F.26) and (F.18) into (F.25).

Appendix G

Generation of Bivariate Innovations

For readers' convenience, the copula-based algorithms in Channouf and L'Ecuyer [21] and Avramidis et al. [9] are provided here for the generation of the bivariate innovation process $(\widehat{\varepsilon}^{(Q)}(t), \widehat{\varepsilon}^{(D)}(t))^\top$. The characteristics of $(\widehat{\varepsilon}^{(Q)}(t), \widehat{\varepsilon}^{(D)}(t))^\top$ are estimated and give as: $\widehat{\mathrm{E}}[\widehat{\varepsilon}^{(Q)}(t)]$, $\widehat{\mathrm{E}}[\widehat{\varepsilon}^{(D)}(t)]$, $\widehat{\mathrm{Var}}[\widehat{\varepsilon}^{(Q)}(t)]$, $\widehat{\mathrm{Var}}[\widehat{\varepsilon}^{(D)}(t)]$ and $\widehat{\mathrm{Cov}}[\widehat{\varepsilon}^{(Q)}(t), \widehat{\varepsilon}^{(D)}(t)]$. The marginal cumulative distribution functions (CDF) for $\widehat{\varepsilon}^{(Q)}(t)$ and $\widehat{\varepsilon}^{(D)}(t))^\top$ are specified as F_Q and F_D, respectively. In this work, the CDF could be normal or generalized Poisson (GP) as discussed in Section 5.1.1, and herein we consider the case where at least one of F_Q and F_D follows GP.

In the notations below, the time index t is omitted for clarity. Let $\mathbf{u} = (u^{(Q)}, u^{(D)})^\top$ be a bivariate normal random variable with

$$\mathrm{E}[\mathbf{u}] = \mathbf{0}, \quad \mathrm{Cov}[\mathbf{u}] = \begin{pmatrix} 1 & \varphi \\ \varphi & 1 \end{pmatrix}.$$

Define $\widehat{\boldsymbol{\varepsilon}}$ as

$$\widehat{\boldsymbol{\varepsilon}} = (\widehat{\varepsilon}^{(Q)}, \widehat{\varepsilon}^{(D)})^\top = (F_Q^{-1}(\Phi(u^{(Q)})), F_D^{-1}(\Phi(u^{(D)})))^\top$$

The Φ denotes the CDF of stand normal distribution. The covariance between $\widehat{\varepsilon}^{(Q)}$ and $\widehat{\varepsilon}^{(D)}$ is given as:

$$\mathrm{Cov}[\widehat{\varepsilon}^{(Q)}, \widehat{\varepsilon}^{(D)}] = \mathrm{E}[\widehat{\varepsilon}^{(Q)}\widehat{\varepsilon}^{(D)}] - \mathrm{E}[\widehat{\varepsilon}^{(Q)}]\mathrm{E}[\widehat{\varepsilon}^{(D)}]$$

where

$$\mathrm{E}[\widehat{\varepsilon}^{(Q)}\widehat{\varepsilon}^{(D)}] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F_Q^{-1}(\Phi(u^{(Q)}))F_D^{-1}(\Phi(u^{(D)}))\phi_{\varphi}(u^{(Q)}, u^{(D)})du^{(Q)}du^{(D)}. \tag{G.1}$$

In (G.1), $\phi_{\varphi}(u^{(Q)}, u^{(D)})$ denotes the probability density function of $\mathbf{u}$. Obviously, the value of $\mathrm{E}[\widehat{\varepsilon}^{(Q)}\widehat{\varepsilon}^{(D)}]$ depends on the choice of φ. Hence at this point, the problem of generating $\widehat{\boldsymbol{\varepsilon}}$ with given characteristics turns into the covariance matching problem: Find the value of φ such that

$$\mathrm{E}[\widehat{\varepsilon}^{(Q)}\widehat{\varepsilon}^{(D)}] = \widehat{\mathrm{Cov}}[\widehat{\varepsilon}^{(Q)}, \widehat{\varepsilon}^{(D)}] + \widehat{\mathrm{E}}[\widehat{\varepsilon}^{(Q)}]\widehat{\mathrm{E}}[\widehat{\varepsilon}^{(D)}]. \tag{G.2}$$

To solve the covariance matching problem in (G.2), the integral of $\mathrm{E}[\widehat{\varepsilon}^{(Q)}\widehat{\varepsilon}^{(D)}]$ in (G.1) is first transformed into a sum of terms, and then a certain numerical algorithm is employed to find $\widehat{\varphi}$ that solves (G.2) approximately [21] and [9].

Given the F_Q, F_D and $\widehat{\mathrm{Cov}}[\widehat{\varepsilon}^{(Q)}, \widehat{\varepsilon}^{(D)}]$, the algorithm for generating samples of bivariate random variables $\{\widehat{\boldsymbol{\varepsilon}}_i; i = 1, 2, ..., N\}$ are summarized as follow:

Step 1 : Use the the formula (19) in Avramidis et al. [9] if both $\widehat{\varepsilon}^{(Q)}$ and $\widehat{\varepsilon}^{(D)}$ follow GP distribution, or (11) in Channouf and L'Ecuyer [21] if one of them follows normal distribution, to find $\widehat{\varphi}$ that solves (G.2) via numerical search (e.g. Newton-Raphson method).

Step 2 : Generate independent bivariate standard normal variables $\{\mathbf{u}_i = (u_i^{(Q)}, u_i^{(D)})^{\top}, i = 1, 2, ..., N\}$, where each $\mathbf{u}_i$ satisfies:

$$\mathrm{E}[\mathbf{u}_i] = \mathbf{0},\;\; \mathrm{Cov}[\mathbf{u}_i] = \begin{pmatrix} 1 & \widehat{\varphi} \\ \widehat{\varphi} & 1 \end{pmatrix}.$$

Many softwares (e.g. Matlab, R) provide efficient algorithms to complete this step.

Step 3 : For each i, obtain $\widehat{\boldsymbol{\varepsilon}}_i = (\widehat{\varepsilon}_i^{(Q)}, \widehat{\varepsilon}_i^{(D)})^\top$ by

$$\widehat{\boldsymbol{\varepsilon}}_i^{(Q)} = F_Q^{-1}(\Phi(z_i^{(Q)})), \;\; \widehat{\varepsilon}^{(D)} = F_D^{-1}(\Phi(z_i^{(D)})).$$

Appendix H

Two-Stage Procedure for Sample Size Determination

For a candidate release plan $\mathbf{x}$, how to determine the number of Monte Carlo simulation (MCS) replications R needed to obtain high-quality estimates of the performance metrics: the expected total cost $\mathrm{E}[TC(\mathbf{x})]$ and the demand fulfill rate $\mathrm{E}[DF(\mathbf{x})]$? Denote a performance metric as $M(\mathbf{x})$ in general, where $M(\mathbf{x})$ can be $TC(\mathbf{x})$ or $DF(\mathbf{x})$. In this work, the two-stage procedure [64, 90] is implemented to determine the value of R so that a desired precision can be achieved for $\widehat{\mathrm{E}}[M(\mathbf{x})]$, that is,

$$\frac{\widehat{\mathrm{Std}}\left[\widehat{\mathrm{E}}[M(\mathbf{x})]\right]}{\widehat{\mathrm{E}}[M(\mathbf{x})]} < \gamma\%$$

where $\widehat{\mathrm{Std}}\left[\widehat{\mathrm{E}}[M(\mathbf{x})]\right]$ is the estimated standard deviation of $\widehat{\mathrm{E}}[M(\mathbf{x})]$, and $\gamma\%$ the target precision level. At the first stage, an initial number of replications R_0 are generated. Denote $\widehat{\mathrm{E}}_0[M(\mathbf{x})]$ as the estimated mean of $M(\mathbf{x})$ and $\widehat{\mathrm{Std}_0}\left[\widehat{\mathrm{E}_0}[M(\mathbf{x})]\right]$ as the estimated standard deviation of $\widehat{\mathrm{E}}_0[M(\mathbf{x})]$ from the R_0 replications. Then the number of replications R that is likely to achieve the target precision level $\gamma\%$ is estimated as

$$R = \lceil \widehat{\mathrm{Std}_0}\left[\widehat{\mathrm{E}_0}[M(\mathbf{x})]\right] / \left(\widehat{\mathrm{E}}_0[M(\mathbf{x})] \times \gamma\%\right) \rceil.$$

At the second stage, $R - R_0$ additional replications are generated and the data collected at both stages are used to estimate the performance metrics.

Appendix I

Simulating Demand Process

The demand process $\mathcal{D}(t)$ is written as:

$$\mathcal{D}(t) = \mathrm{E}[\mathcal{D}(t)] + v(t), \quad t = 1, 2, ..., H$$

where $\mathrm{E}[\mathcal{D}(t)]$ is pre-given, and $v(t)$ is a stochastic process representing the deviation of $\mathcal{D}(t)$ from its mean $\mathrm{E}[\mathcal{D}(t)]$. Denote the demand realization times as

$$\{\tau_1,\ \tau_2, ..., \tau_U\}; \quad 1 \leq \tau_1 < \tau_2 < ... < \tau_U \leq H,$$

with U being the number of demand realizations over the planning horizon. $\mathrm{E}[\mathcal{D}(t)]$ and $v(t)$ are set as 0 at the time points when no demand is realized, that is

$$\mathrm{E}[\mathcal{D}(t)] = 0,\ v(t) = 0,\ t \notin \{\tau_1,\ \tau_2, ..., \tau_U\}.$$

In this work, $v(\tau_i)$ is modeled by a AR-GARCH process [13, 31], which is widely used in modeling and forecasting demand [36, 83, 85, 86]. The AR-GARCH model is given as

$$v(\tau_i) = \xi v(\tau_{i-1}) + \epsilon(\tau_i)\sigma(\tau_i) \tag{I.1}$$

$$\sigma(\tau_i)^2 = \phi_0 + \phi_1(\epsilon(\tau_{i-1})\sigma(\tau_{i-1}))^2 + \psi\sigma(\tau_{i-1})^2 \tag{I.2}$$

where $\epsilon(\tau_i)$ is Gaussian white noise with unit variance, and ξ, ϕ_0, ϕ_1 and ψ are given model parameters, which determines the variance of the demand process. The demand process can be simulated by (I.1) and (I.2).

References

[1] E. Albey, U. Bilge, and R. Uzsoy. An exploratory study of disaggregated clearing functions for multiple product single machine production environments. *EP Fitts Department of Industrial and Systems Engineering*, 2011.

[2] E. Albey, Ü. Bilge, and R. Uzsoy. An exploratory study of disaggregated clearing functions for production systems with multiple products. *International Journal of Production Research*, 52(18):5301–5322, 2014.

[3] A. A. Alzaid and M. Al-Osh. An integer-valued pth-order autoregressive structure (INAR(p)) process. *Journal of Applied Probability*, pages 314–324, 1990.

[4] B. Ankenman, B. L. Nelson, and J. Staum. Stochastic kriging for simulation metamodeling. *Operations Research*, 58(2):371–382, 2010.

[5] T. Aouam and R. Uzsoy. Chance-constraint-based heuristics for production planning in the face of stochastic demand and workload-dependent lead times. In *Decision Policies for Production Networks*, pages 173–208. Springer, 2012.

[6] T. Aouam and R. Uzsoy. Zero-order production planning models with stochastic demand and workload-dependent lead times. *International Journal of Production Research*, (ahead-of-print):1–19, 2014.

[7] J. M. Asmundsson, R. L. Rardin, and Re. Uzsoy. Tractable nonlinear production planning models for semiconductor wafer fabrication facilities. *IEEE Transactions on Semiconductor Manufacturing*, 19(1):95–111, 2006.

[8] J. M. Asmundsson, R. L. Rardin, and Re. Uzsoy. Production planning models for resources subject to congestion. *Naval Research Logistics*, 56:142–157, 2009.

[9] A. N. Avramidis, N. Channouf, and P. L'Ecuyer. Efficient correlation matching for fitting discrete multivariate distributions with arbitrary marginals and normal-copula dependence. *INFORMS Journal on Computing*, 21(1):88–106, 2009.

[10] R. Azrak and G. Melard. Asymptotic properties of quasi-maximum likelihood estimators for arma models with time-dependent coefficients. *Statistical Inference for Stochastic Processes*, 9(3):279–330, 2006.

[11] D. Bertsimas and G. Mourtzinou. Transient laws of non-stationary queueing systems and their applications. *Queueing Systems*, 25(1-4):115–155, 1997.

[12] P. J. Billington, J. O. McClain, and L. J. Thomas. Mathematical programming approaches to capacity-constrained MRP systems: review, formulation and problem reduction. *Management Science*, 29(10):1126–1141, 1983.

[13] T. Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3):307–327, 1986.

[14] G. E. Box, G. M. Jenkins, and G. C. Reinsel. *Time series analysis: forecasting and control*. John Wiley & Sons, 2013.

[15] K. Brännäs. Explanatory variables in the AR(1) count data model. *Umeå Economic Studies*, 381, 1995.

[16] P. J. Brockwell and R. A. Davis. *Introduction to time series and forecasting*, volume 1. Taylor & Francis, 2002.

[17] P. J. Brockwell and R. A. Davis. *Time series: theory and methods*. Springer, 2009.

[18] M. D. Byrne and M. A. Bakir. Production planning using a hybrid simulationanalytical approach. *International Journal of Production Economics*, 59(1):305–311, 1999.

[19] M. D. Byrne and M. M. Hossain. Production planning: an improved hybrid approach. *International Journal of Production Economics*, 93–94:225–229, 2005.

[20] Y. Cai, J. Huang, Y. Tang, and G. Zhou. A simulation method for finite non-stationary time series. *Journal of Statistical Computation and Simulation*, 84(7):1563–1579, 2014.

[21] N. Channouf and P. L'Ecuyer. Fitting a normal copula for a multivariate distribution with both discrete and continuous marginals. In *Proceedings of the Winter Simulation Conference*, pages 352–358. Winter Simulation Conference, 2009.

[22] H. B. Chen and A. Mandelbaum. Hierarchical modeling of stochastic networks, part i: fluid models. In *Stochastic modeling and analysis of manufacturing systems*, pages 47–105. Springer, New York, 1994.

[23] C. Chesneau and M. Kachour. A parametric study for the first-order signed integer-valued autoregressive process. *Journal of Statistical Theory and Practice*, 6(4):760–782, 2012.

[24] P. C. Consul. *Generalized Poisson distributions: properties and applications.* M. Dekker, New York and Basel, 1989.

[25] R. A. Davis, W. T. M. Dunsmuir, and S. B. Streett. Observation-driven models for Poisson counts. *Biometrika*, 90(4):777–790, 2003.

[26] K. Deb. *Multi-objective optimization using evolutionary algorithms*, volume 16. John Wiley & Sons, 2001.

[27] J. del Castillo and M. Pérez-Casany. Overdispersed and underdispersed Poisson generalizations. *Journal of Statistical Planning and Inference*, 134(2):486–500, 2005.

[28] J. Du and Y. Li. The integer-value autoregressive (inar(p)) model. *Journal of Time Series Analysis*, 12(2):129–142, 1991.

[29] B. Efron. Double exponential families and their use in generalized linear regression. *Journal of the American Statistical Association*, 81(395):709–721, 1986.

[30] Neal P. Enciso-Mora, V. and T. S. Rao. Integer valued AR processes with explanatory variables. *Sankhyā: The Indian Journal of Statistics, Series B*, pages 248–263, 2009.

[31] R. F. Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econometrica: Journal of the Econometric Society*, pages 987–1007, 1982.

[32] G. D. Eppen and R. K. Martin. Determining safety stock in the presence of stochastic lead time and demand. *Management Science*, 34(11):1380–1390, 1988.

[33] W. Fei and L. Bai. Time-varying parameter auto-regressive models for autocovariance nonstationary time series. *Science in China Series A: Mathematics*, 52(3):577–584, 2009.

[34] Wanchun Fei and Lun Bai. Auto-regressive models of non-stationary time series with finite length. *Tsinghua Science & Technology*, 10(2):162–168, 2005.

[35] K. Fokianos. Count time series models. *Handbook in Statistics: Time Series Methods and Applications*, 30:315–348, 2012.

[36] Contreras J. Van Akkeren M. Garcia, R. C. and J. B. C. Garcia. A GARCH forecasting model to predict day-ahead electricity prices. *Power Systems, IEEE Transactions on*, 20(2):867–874, 2005.

[37] G. K. Grunwald, R. J. Hyndman, L. Tedesco, and R. L. Tweedie. Theory & methods: non-gaussian conditional linear AR(1) models. *Australian & New Zealand Journal of Statistics*, 42(4):479–495, 2000.

[38] S. T. Hackman and R. C. Leachman. A general framework for modeling production. *Management Science*, 35(4):478–495, 1989.

[39] S. Haeussler and H. Missbauer. Empirical validation of metamodels of work centers in order release planning. *International Journal of Production Economics*, 149:102–116, 2014.

[40] J. H. Heizer and B. Render. *Operations management*. Pearson Prentice Hall, Upper Saddle River, NJ, 2008.

[41] J. L. Higle and K. G. Kempf. Production planning under supply and demand uncertainty: A stochastic programming approach. In G. Infanger, editor, *Stochastic Programming: The State of the Art*, pages 297 – 315. Springer, Berlin, 2010.

[42] L. J. Hong and B. L. Nelson. Discrete optimization via simulation using compass. *Operations Research*, 54(1):115–129, 2006.

[43] Y. F. Hung and M. C. Hou. A production planning approach based on iterations of linear programming optimization and flow time prediction. *Journal of the Chinese Institute of Industrial Engineers*, 18(3):55–67, 2001.

[44] Y. F. Hung and R. C. Leachman. A production planning methodology for semiconductor manufacturing based on iterative simulation and linear programming calculations. *IEEE Transactions on Semiconductor Manufacturing*, 9(2):257–269, 1996.

[45] D. F. Irdem, N. B. Kacar, and R. Uzsoy. An experimental study of an iterative simulation-optimization algorithm for production planning. In S. J. Mason, R. Hill, L. Moench, and O. Rose, editors, *Proceedings of the Winter Simulation Conference*, pages 2176–2184, Miami FL, 2008.

[46] D. F. Irdem, N. B. Kacar, and R. Uzsoy. An exploratory analysis of two iterative linear programming-simulation approaches for production planning. *IEEE Transactions on Semiconductor Manufacturing*, 23(3):442–455, 2010.

[47] H. Joe. Time series models with univariate margins in the convolution-closed infinitely divisible class. *Journal of Applied Probability*, pages 664–677, 1996.

[48] L. A. Johnson and D. C. Montgomery. *Operations research in production planning, scheduling, and inventory control.* Wiley, New York, 1974.

[49] R. C. Jung, M. Kukuk, and R. Liesenfeld. Time series of count data: modeling, estimation and diagnostics. *Computational Statistics & Data Analysis*, 51(4):2350–2364, 2006.

[50] R. C. Jung and A. R. Tremayne. Convolution-closed models for count time series with applications. *Journal of Time Series Analysis*, 32(3):268–280, 2011.

[51] R. C. Jung and A. R. Tremayne. Useful models for time series of counts or simply wrong ones? *AStA Advances in Statistical Analysis*, 95(1):59–91, 2011.

[52] N. B. Kacar, D. F. Irdem, and R. Uzsoy. An experimental comparison of production planning using clearing functions and iterative linear programming-simulation algorithms. *IEEE Transactions on Semiconductor Manufacturing*, 25(1):104–117, 2012.

[53] N. B. Kacar and R. Uzsoy. Estimating clearing functions from simulation data. In *Proceedings of the Winter Simulation Conference*, pages 1699–1710. Winter Simulation Conference, 2010.

[54] N. B. Kacar and R. Uzsoy. A comparison of multiple linear regression approaches for fitting clearing functions to empirical data. *International Journal of Production Research*, 52(11):3164–3184, 2014.

[55] M. Kachour and L. Truquet. A p-order signed integer-valued autoregressive (SINAR(p)) model. *Journal of Time Series Analysis*, 32(3):223–236, 2011.

[56] M. Kachour and J. Yao. First-order rounded integer-valued autoregressive (RINAR(1)) process. *Journal of Time Series Analysis*, 30(4):417–448, 2009.

[57] U. S. Karmarkar. Capacity loading and release planning with work-in-progress (wip) and lead-times. *Journal of Manufacturing and Operations Management*, 2:105–123, 1989.

[58] D. Kayton, T. Teyner, C. Schwartz, and R. Uzsoy. Focusing maintenance improvement efforts in a wafer fabrication facility operating under the theory of constraints. *Production and Inventory Management Journal*, 38:51–57, 1997.

[59] F. P. Kelly, S. Zachary, and I. Ziedins. Stochastic networks: theory and applications, 1996.

[60] B. Kim and S. Kim. Extended model for a hybrid production planning approach. *International Journal of Production Economics*, 73(2):165–173, 2001.

[61] H. Kim and Y. Park. A non-stationary integer-valued autoregressive model. *Statistical Papers*, 49(3):485–502, 2008.

[62] A. M. Law and W. D. Kelton. *Simulation modeling and analysis.* McGraw-Hill, New York, third edition, 2000.

[63] J. Liu. *Simulation-based transfer function modeling approach for responsive production planning.* PhD thesis, West Virginia University, 2011.

[64] J. G. Liu, C. H. Li, F. Yang, H. Wan, and R. Uzsoy. Production planning for semiconductor manufacturing via simulation optimization. In S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, editors, *Proceedings of the Winter Simulation Conference*, pages 3617–3627, New York, 2011.

[65] L. Ljung. *System identification: theory for the user, PTR Prentice Hall Information and System Sciences Series.* Prentice Hall, New Jersey, second edition, 1999.

[66] A. Mandelbaum and W. A. Massey. Strong approximations for time-dependent queues. *Mathematics of Operations Research*, 20(1):33–64, 1995.

[67] F. J. Massey Jr. The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American Statistical Association*, 46(253):68–78, 1951.

[68] E. McKenzie. Discrete variate time series. *Handbook of Statistics*, 21:573–606, 2003.

[69] H. Missbauer. Models of the transient behaviour of production units to optimize the aggregate material flow. *International Journal of Production Economics*, 118(2):387–397, 2009.

[70] H. Missbauer. Order release planning with clearing functions: a queueing-theoretical analysis of the clearing function concept. *Intational Journal of Production Economics*, 131:399–406, 2011.

[71] H. Missbauer and R. Uzsoy. Optimization models of production planning problems. In K. G. Kempf, P. Keskinocak, and R. Uzsoy, editors, *Planning Production and Inventories in the Extended Enterprise: A State of the Art Handbook*, pages 437 – 507. Springer, New York, 2011.

[72] Y. Narahari, N. Hemachandra, and M. S. Gaur. Transient analysis of multiclass manufacturing systems with priority scheduling. *Computers & Operations Research*, 24(5):387–398, 1997.

[73] S. Orcun, R. Uzsoy, and K. G. Kempf. An integrated production planning model with load-dependent lead-times and safety stocks. *Computers & Chemical Engineering*, 33(12):2159–2163, 2009.

[74] B. Pfaff. Non-stationary time series. *Analysis of Integrated and Cointegrated Time Series with R*, pages 53–71, 2008.

[75] Y. Pochet and L. A. Wolsey. *Production planning by mixed integer programming.* Springer, 2006.

[76] T. S. Rao. The fitting of non-stationary time-series models with time-dependent parameters. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 312–322, 1970.

[77] A. Ravindran, K. G. Kempf, and R. Uzsoy. Production planning with load dependent lead times and safety stocks for a single product. *International Journal of Planning and Scheduling*, 1(1):58–89, 2011.

[78] G. Riãno. *Transient behavior of stochastic networks: application to production planning with load-dependent lead times.* PhD thesis, Georgia Institute of Technology, 2003.

[79] B. D. Ripley. *Stochastic simulation*, volume 316. John Wiley & Sons, 2009.

[80] Borle S. Sellers, K. F. and G. Shmueli. The COM-Poisson model for count data: a survey of methods and applications. *Applied Stochastic Models in Business and Industry*, 28(2):104–116, 2012.

[81] J. G. Shanthikumar, Ding S. W., and M. T. Zhang. Queueing theory for semiconductor manufacturing systems: a survey and open problems. *IEEE Transactions on Automation Science and Engineering*, 4(4):513–522, 2007.

[82] S. S. Shapiro and M. B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, pages 591–611, 1965.

[83] H. Song and G. Li. Tourism demand modelling and forecastinga review of recent research. *Tourism Management*, 29(2):203–220, 2008.

[84] F. W. Steutel and K. Van Harn. Discrete analogues of self-decomposability and stability. *The Annals of Probability*, 7(5):893–899, 1979.

[85] J. W. Taylor and R. Buizza. A comparison of temperature density forecasts from GARCH and atmospheric models. *Journal of Forecasting*, 23(5):337–355, 2004.

[86] McSharry P. E. Taylor, J. W. and R. Buizza. Wind power density forecasting using ensemble predictions and time series models. *IEEE Transactions on Energy Conversion*, 24(3):775–782, 2009.

[87] C. H. Weiß. Thinning operations for modeling time series of counts-a survey. *AStA Advances in Statistical Analysis*, 92(3):319–341, 2008.

[88] C. H. Weiß. Integer-valued autoregressive models for counts showing underdispersion. *Journal of Applied Statistics*, 40(9):1931–1948, 2013.

[89] M. B. Wilk and R. Gnanadesikan. Probability plotting methods for the analysis for the analysis of data. *Biometrika*, 55(1):1–17, 1968.

[90] F. Yang and J. Liu. Simulation-based transfer function modeling for transient analysis of general queueing systems. *European Journal of Operational Research*, 223:150–166, 2012.

[91] C. A. Yano and R. C. Carlson. Safety stocks for assembly systems with fixed production intervals. *Ann Arbor*, 1001:48109–2117, 1987.

[92] S. L. Zeger. A regression model for time series of counts. *Biometrika*, 75(4):621–629, 1988.

[93] S. L. Zeger and B. Qaqish. Markov regression models for time series: a quasi-likelihood approach. *Biometrics*, pages 1019–1031, 1988.

[94] F. Zhu. Modeling overdispersed or underdispersed count data with generalized Poisson integer-valued GARCH models. *Journal of Mathematical Analysis and Applications*, 389(1):58–71, 2012.