Modelling and Stochastic Simulation of Supply Chains

Sohail Suleman Zaveri
201681438

Supervised by Dr. Nadhir Ben Rached (Department of Mathematics)
and Hamish Steptoe (Met Office)

Submitted in accordance with the requirements for the
module MATH5872M: Dissertation in Data Science and Analitics
as part of the degree of

Master of Science in Data Science and Analytics

The University of Leeds, School of Mathematics

September 2023

The candidate confirms that the work submitted is his/her own and that appropriate
credit has been given where reference has been made to the work of others.

UNIVERSITY OF LEEDS

School of Mathematics

FACULTY OF ENGINEERING AND PHYSICAL SCIENCES

Academic integrity statement

I am aware that the University defines plagiarism as presenting someone else's work, in whole or in part, as your own. Work means any intellectual output, and typically includes text, data, images, sound or performance.

I promise that in the attached submission I have not presented anyone else's work, in whole or in part, as my own and I have not colluded with others in the preparation of this work. Where I have taken advantage of the work of others, I have given full acknowledgement. I have not resubmitted my own work or part thereof without specific written permission to do so from the University staff concerned when any of this work has been or is being submitted for marks or credits even if in a different module or for a different qualification or completed prior to entry to the University. I have read and understood the University's published rules on plagiarism and also any more detailed rules specified at School or module level. I know that if I commit plagiarism I can be expelled from the University and that it is my responsibility to be aware of the University's regulations on plagiarism and their importance.

I re-confirm my consent to the University copying and distributing any or all of my work in any form and using third parties (who may be based outside the EU/EEA) to monitor breaches of regulations, to verify whether my work contains plagiarised material, and for quality assurance purposes. I confirm that I have declared all mitigating circumstances that may be relevant to the assessment of this piece of work and that I wish to have taken into account. I am aware of the University's policy on mitigation and the School's procedures for the submission of statements and evidence of mitigation. I am aware of the penalties imposed for the late submission of coursework.

Name Sohail Suleman Zaveri

Student ID 201681438

Abstract

Write your abstract here. It should be no longer than one page.

Contents

List of Figures

List of Tables

Chapter 1

Introduction to Supply Chains

A *supply chain* **?** is a network of all the parties involved in production, transportation, storage, distribution and delivery of products in order to fulfil a customer's request. Other than manufacturers and suppliers, a supply chain also includes warehouses, transporters, distributors, retailers and the customers themselves. A supply chain also includes the activities and functions that are aimed towards fulfilling a customer's demands. These functions include distribution, finance, quality assurance, information exchange and transportation.

We present a simple example of a supply chain that describes the functions of various stages when a customer buys a smartphone from a Walmart store.

Let us consider a customer purchasing a smartphone from Walmart. The supply chain that kicks-in to fulfil the customer's request includes - the customer, Walmart's store staff, Walmart's distribution centers, smartphone manufacturer, their suppliers, transportation partners and third-party organisations. The supply chains starts with the customer visiting Walmart to buy a smartphone. Walmart's store staff stocks its shelves from the store's inventory, that in turn is supplied from Walmart's distribution centers. Supplying goods requires involvement of transportation vehicles that might have been supplied from a third party. The distribution centers are stocked by supplies from smartphone manufacturers, that themselves use supplies from lower-tier suppliers. For example, the manufactures might depend on a supplier for its semiconductor demands and another supplier for its packaging needs. This network of stages involves information flow amongst parties such as - i) point-of-sales information about the purchase that is registered at Walmart store, ii) data on available stocks in the store's inventory that is communicated with Walmart's distribution centers, iii) sales information and additional orders of smartphones placed by Walmart to smartphone manufacturers, just to name a few.

Several stages that are involved in a typical supply chain, as described in **?**, include the customers, retailers, distributors, manufacturers and raw material suppliers.

A manufacturer usually depends on more than one suppliers, and similarly other stages of a supply chain depend on more than one parties for proper functioning of the supply chain. Figure 1.1 depicts a supply chain with eight processes, where process 3 depends on supplies from supply 1 and process 1. Similarly, process 5 depends on output materials from processes

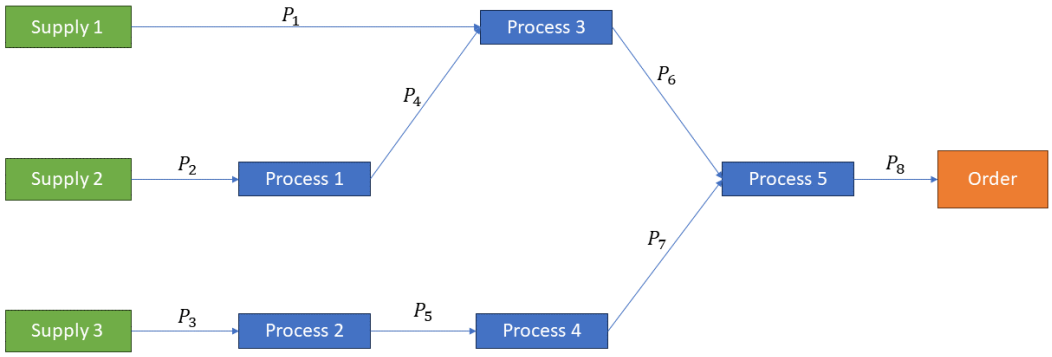


Figure 1.1: A supply chain with 3 supplies and 5 processes.

3 and 4.

1.1 Risk and disruption in supply chains

Supply chains across the globes faced one of the biggest disruptions in the form of COVID-19 pandemic in 2020-2021. Supply chains at local and global levels were effected drastically due to the near-total reduction in public mobility and ocean freight. Reduction in workforce at manufacturing plants, ports and road transportation posed challenges to supply chains at almost all levels. Many countries still face economic challenges as an aftermath of disrupted and restructured supply chains. Given the far-reaching impact of the COVID-19 pandemic, which led to substantial reductions in public mobility and disrupted various aspects of supply chains, it becomes crucial to discuss the risks and underlying causes of such disruptions during the design and implementation of supply chains.

With rising global temperatures, extreme weather events are posing higher and higher risks to the smooth functioning of global supply chains. High level of inter-dependence between countries for raw materials and manufactured products implies that even a single disruption caused by extreme weather can hamper global supply chains, and hence lead to serious economic consequences.

As global supply chains become increasingly complex, they are susceptible to multiple points of failure, rendering them more vulnerable and less resilient. Some of the risk factors, as discussed in **?**, are categorised in Table 1.1:

Category	Risk Drivers
Delays	High capacity utilization at supply source, inflexibility of supply source, poor quality or yield at supply source
Disruptions	Pandemics, war, natural disaster, supplier bankruptcy, labor disputes
Systems Risk	Information infrastructure breakdown, system integration or extent of systems being networked
Forecast Risk	Inaccurate forecasts due to small customer base, seasonality, product variety, information distortion
Procurement Risk	Exchange-rate risk, price of inputs, fraction purchased from a single source, industry-wide capacity utilization
Receivables Risk	Number of customers, financial strength of customers
Inventory Risk	Rate of product obsolescence, inventory holding cost, product value, demand and supply uncertainty
Capacity Risk	Cost of capacity, capacity flexibility

Table 1.1: Categorization of risk factors in supply chains

1.2 Resilience of supply chains

With good network designs, supply chains can mitigate many risk factors mentioned in Table 1.1. According to D. Mavatoor in **?**, "The concept of resilience shouldn't assume that you won't fail, but rather that you should be able to get back up fast." Flexibility of networks, amongst others, plays a crucial role in ensuring resilience of supply chains. An example of the effectiveness of flexible networks is the levels of impact a supplier disruption had on two companies, Nokia and Ericsson in 2000. Supply chains of Nokia and Ericsson were disrupted when a plant owned by Royal Philips Electronics caught fire in Albuquerque, New Mexico in 2000. Nokia bounced back quickly because of presence of multiple backups in their network of suppliers, and faced minimal losses. On the other hand, Ericsson lost $\$400$ million in revenues as it did not have a backup in their network, and hence could not bounce back fast.

Mitigation strategies come at a price and their own set of risks. Having multiple suppliers in the network increases cost of the supply chain due to increase in transportation, coordination, management and used resources. Having higher inventory incurs cost on storage, maintenance and raises risk of obsolescence. As discussed in **?**, the trade-off in inventory control is between *producing*, which leads to increased inventory costs and management costs, and *idling*, which can lead to stock-outs and unsatisfied demands. Hence, customised mitigation strategies should be incorporated in designs of supply chains that achieve a trade-off between the risk mitigated and the incurred cost. Some of the tailoring strategies, as discussed in **?** are mentioned in table 1.2.

Risk Mitigation Strategy	Tailored Strategies
Increase capacity	Focus on low-cost, decentralized capacity for predictable demand. Build centralized capacity for unpredictable demand. Increase decentralization as the cost of capacity drops.
Get redundant suppliers	More redundant supply for high-volume products, less redundancy for low-volume products. Centralize redundancy for low-volume products in a few flexible suppliers.
Increase responsiveness	Favor cost over responsiveness for commodity products. Favor responsiveness over cost for short-life cycle products.
Increase inventory	Decentralize inventory of predictable, lower value products. Centralize inventory of less predictable, higher value products.
Increase flexibility	Favor cost over flexibility for predictable, high-volume products. Favor flexibility for unpredictable, low-volume products. Centralize flexibility in a few locations if it is expensive.
Pool or aggregate demand	Increase aggregation as unpredictability grows.
Increase source capability	Prefer capability over cost for high-value, high-risk products. Favor cost over capability for low-value commodity products. Centralize high capability in a flexible source if possible.

Table 1.2: Tailored strategies for risk mitigation

1.3 Types of processes in supply chains

The processes in supply chains can be categorised based on whether they operate in response to a customer's order, or in anticipation of customer orders **?**. *Push processes* are executed in anticipation of customer orders, whereas *pull processes* are triggered following a customer's order.

1.3.1 Push Processes

Push processes are executed in anticipation of customer orders. They are based on forecasts of product demands in the future, and hence, are also referred to as *speculative processes* because they operate in response to speculated demand. An example of a push process is a bakery producing a variety of bread and pastries early in the morning and pushing them onto store shelves for customers to purchase. Upstream processes, such as production pf raw materials typically use push models based on forecasts of company demands.

1.3.2 Pull Processes

Pull processes are initiated in response to a customer order. These are also known are *reactive processes* as they react to an actual demand of a product. An example of a pull process is a company manufacturing products only when a customer order is received, ensuring that production is triggered by actual demand. Processes closer to end-customers, such as retailers and

e-commercial websites use pull model as they operate only when a customer order is placed.

1.3.3 Hybrid Systems

In a hybrid system, some processes operate on a push model, whereas other processes operate on a pull model. Using hybrid systems enable companies to tailor their strategies to meet their limitations and maximise customer satisfaction at the same time. Hybrid systems offer the potential to optimise inventory levels, enhance customer satisfaction, reduce lead times, and improve overall supply chain performance.

Chapter 2

Chemical Reaction Modelling

Continuous-time Markov chains (CTMCs) are stochastic processes that follow the Markov property in which the system makes transitions in continuous time. Unlike discrete-time Markov chains where transitions can only take place at discrete time steps, CTMCs can transition through states at any point in time.

Stochastic Reaction Networks (SRNs) are a class of CTMCs that are used to model evolution of chemical systems where molecules of different chemical species can undergo a finite set of reactions **?**. SRNs are particularly useful when dealing with systems with small number of molecules as it helps in incorporating the inherent randomness of phase space of molecules in the evolution of the system.

In this chapter, we define CTMCs, SRNs, and discuss algorithms such as *Stochastic Simulation Algorithm* (SSA), *Next Reaction Method* (NRM), *Modified Next Reaction Method* (MNRM) and *tau-leap method*, that can be used to simulate SRN trajectories. SSA, NRM and MNRM are considered to be exact as the paths simulated by these methods follow correct statistical distributions. Tau-leap methods are used to simulate SRN paths using discrete time steps, in order to address some of the computational and performance related shortcomings of exact methods.

2.1 Continuous Time Markov Chains

Let $\mathbf{X} = \{X(t), t \in [0, \infty)\}$ be a continuous-time stochastic process with a countable state space $\mathbf{S}$. $\mathbf{X}$ is called a continuous-time Markov chain **?** if for any set of time indices $0 \leq t_1 < t_2 < \ldots < t_{s-1} < t_s < t$ and corresponding states $c_1, c_2, \ldots, c_{s-1}, c_s, c_t \in \mathbf{S}$ such that $P(X(t_1) = c_1, X(t_2) = c_2, \ldots, X(t_s) = c_s) > 0$, the following is true:

$$P(X(t) = c_t | X(t_1) = c_1, \ldots, X(t_s) = c_s) = P(X(t) = c_t | X(t_s) = c_s). \tag{2.1}$$

The property (2.1) is also called the Markov Property. We see from (2.1) that the probability of X being in a state c_t at time t is only conditioned on the state $X(t_s)$ at time index t_s, and does not depend on the states of the process in time indices $t_1, t_2, \cdots, t_{s-1}$.

2.2 Stochastic Reaction Networks

An SRN $\mathbf{X}$ is a type of CTMC that describes evolution of a homogeneous chemical mixture of molecules in continuous time.

Given an underlying probability space $(\Omega, \mathcal{F}, P)$, $X : [0,T] \times \Omega \to \mathbb{Z}^N_{\geq 0}$ describes the time-evolution of a system of molecules belonging to N different species $(S_1, S_2, \ldots, S_N)$ in continuous time $t \in [0,T]$. Elements in $X(t) = (x_1(t), x_1(t), \ldots, x_N(t))$ indicate the abundances of the N species of molecules at time t. The molecules can undergo a finite set of reactions $\mathbf{R} = (\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_M)$, where a reaction $\mathcal{R}_j \in \mathbf{R}$ consists of a pair (ν_j, a_j). Each element in $\nu_j = (\nu_{j,1}, \nu_{j,2}, \ldots, \nu_{j,N}), 1 \leq j \leq, M$ corresponds to the change in the number of molecules of that particular species when the j^th reaction fires in the mixture of molecules. The vectors ν_j is also known as *stoichiometric vectors*. The *propensity* $a_j : \mathbb{Z}^N_{\geq 0} \to \mathbb{R}_{\geq 0}$ of a reaction $\mathcal{R}_j$ depends on the state $X(t)$ at time t and describes the reaction rate of $\mathcal{R}_j$ as follows:

$$P(X(t+\Delta t) = x + \nu_j | X(t) = x) = a_j(x)\Delta t + o(\Delta t). \tag{2.2}$$

A larger value of $a_j(X(t))$ at time t indicates faster rate of reaction $\mathcal{R}_j$ at time t. Probability of firing reaction $\mathcal{R}_j$ in interval $(t, t+\Delta t]$ increases as $a_j(X(t))$ increases.

The probability that no reaction fires in the time interval $(t+\Delta t]$ in the SRN $\mathbf{X}$ is given by:

$$P(X(t+\Delta t) = x | X(t) = x) = 1 - \sum_{j=1}^{M} a_j(x)\Delta t + o(\Delta t). \tag{2.3}$$

Defining a new quantity $a_0(x) := \sum_{j=1}^{M} a_j(x)$, (2.3) can be rewritten as:

$$P(X(t+\Delta t) = x | X(t) = x) = 1 - a_0(x)\Delta t + o(\Delta t). \tag{2.4}$$

Given $X(t) = x$ at time t, it can be derived from (2.4) that the time to fire next reaction follows an exponential distribution with parameter $a_0(x)$.

The stochastic mass-action kinetics principle can be represented by the diagram given below for some reaction $\mathcal{R}_j \in \mathbf{R}$:

$$\alpha_{j,1}S_1 + \ldots + \alpha_{j,n}S_n \to \beta_{j,1}S_1 + \ldots + \beta_{j,n}S_n. \tag{2.5}$$

As implied by (2.5), when a reaction $\mathcal{R}_j$ is fired, $\alpha_{j,1}, \cdots, \alpha_{j,N}$ molecules of the corresponding species $S_1, \cdots, S_N$ are consumed and $\beta_{j,1}, \cdots, \beta_{j,N}$ molecules are produced. The stoichiometric vector $\nu_j := (\beta_{j,1} - \alpha_{j,1}, \ldots, \beta_{j,n} - \alpha_{j,n})$ can contain both positive and negative integers, and hence $\nu_j \in \mathbb{Z}$.

Given a positive reaction constant c_j and state $X(t) = x = (x_1, \ldots, x_N)$ at time t, the

propensity function $a_j(x)$ of reaction $\mathcal{R}_j$ as derived from mass-action kinetics is given by:

$$a_j(x) = c_j \prod_{i=1}^{N} \frac{x_i!}{(x_i - \alpha_{j,i})!} \mathbf{1}_{\alpha_{j,i} \leq x_i} \tag{2.6}$$

where $\alpha_{j,i}$ is the number of S_i molecules consumed when reaction R_j is fired and $\mathbf{1}_{\alpha_{j,i} \leq x_i}$ is an indicator function that can be described as below:

$$\mathbf{1}_{\mathbf{A}}(x) = \begin{cases} 1, & \text{if } x \in \mathbf{A}, \\ 0, & \text{otherwise.} \end{cases}$$

In the remaining literature, (2.6) will be used to calculate propensities of reactions given the state $X(t)$ at time t.

Random Time Change Representation

Let x_0 be the initial state of an SRN $\{X(t), t \geq 0\}$ with reaction channels $(\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_M)$. The random time-change representation of $\mathbf{X}$ as shown by T.G. Kurtz **???** is given as:

$$X(t) = x_0 + \sum_{j=1}^{m} \nu_j Y_j \left(\int_0^t a_i(X(s)) ds \right), \tag{2.7}$$

where $Y_j : \mathbb{R}_{\geq 0} \times \Omega \to \mathbb{Z}_{\geq 0}$ are independent unit-rate Poisson processes. The representation (2.7) models the path of the process $\{X(t), t \geq 0\}$ in continuous time as a combination of paths of M independent unit-rate Poisson processes. Each reaction is represented by a unit-rate Poisson process, and for $1 \leq j \leq M$, the value $\int_0^t a_j(X(s)) ds$ is called the internal time of Y_j.

Later in the chapter, we will see how *Next Reaction Method* (NRM) and *Modified Next Reaction Method* (MNRM) make use of (7) to simulate exact paths of SRNs by sampling next-reation times and next-reaction internal times respectively.

Example - Gene transcription and translation

For illustration, we look at an example of an SRN that models gene transcription and translation. More details on this model can be found in **?**. The SRN, say $X(t), t \geq 0$, consists of three species (M, P, D) and five reactions as shown below:

- $\emptyset \to M$, Transcription of a gene into mRNA.
- $M \to M + P$, Translation of mRNA into proteins.
- $P + P \to D$, Production of stable *Dimers* from proteins.
- $M \to \emptyset, P \to \emptyset$, Degradation of mRNA and proteins respectively.

From the above reactions, we can construct ν_j for each reaction as follows:

$$\nu_1 = \begin{bmatrix}1\\0\\0\end{bmatrix}, \quad \nu_2 = \begin{bmatrix}0\\1\\0\end{bmatrix}, \quad \nu_3 = \begin{bmatrix}0\\-2\\1\end{bmatrix}, \quad \nu_4 = \begin{bmatrix}-1\\0\\0\end{bmatrix}, \quad \nu_5 = \begin{bmatrix}0\\-1\\0\end{bmatrix}.$$

The propensity functions of the five reactions as functions of the state $X(t) = (m(t), p(t), d(t))^T$ at some time t are given by:

$$a_1(X(t)) = 25,\ a_2(X(t)) = 10^3 m(t),\ a_3(X(t)) = 10^{-3} p(t)(p(t)-1),$$
$$a_4(X(t)) = 0.1 m(t) \text{ and } a_5(X(t)) = p(t).$$

If the initial state $X(0) = x_0$, the process can be modeled with Kurtz' random time change representation (2.7) as:

$$\begin{aligned} X(t) = x_0 &+ Y_1\left(\int_0^t 25ds\right)\begin{bmatrix}1\\0\\0\end{bmatrix} + Y_2\left(\int_0^t 10^3 m(s)ds\right)\begin{bmatrix}0\\1\\0\end{bmatrix} \\ &+ Y_3\left(\int_0^t 10^{-3}p(s)(p(s)-1)ds\right)\begin{bmatrix}0\\-2\\1\end{bmatrix} + Y_4\left(\int_0^t 0.1 m(s)ds\right)\begin{bmatrix}-1\\0\\0\end{bmatrix} \\ &+ Y_5\left(\int_0^t p(s)ds\right)\begin{bmatrix}0\\-1\\0\end{bmatrix} \end{aligned}$$

Where Y_js are unit-rate Poisson processes with internal times given as $\int_0^t a_j(X(s))ds$.

2.3 Simulation of SRNs

Analytical and numerical solutions to differential equations involving SRNs are usually infeasible. Some challenges met while finding analytical solutions for an SRN, lets say $\mathbf{X}$, include: i) non-linearity of propensity functions a_js w.r.t. states of $\mathbf{X}$, ii) large number of state space of $X(t), t \geq 0$, to name a few. For this reason, simulating paths of SRNs using Monte Carlo method can help us circumvent these challenges and find approximate values of several quantities of interest including but not limited to time taken by process $X(t), t \geq 0$ to reach a specified subset of its state space and expected value of an observable g at some time T, i.e. $E[g(X(T))]$.

Exact Algorithms

Exact algorithms are methods that can be used to simulate paths of an SRN $X(t), t \geq 0$ that obey the probabilities described by (2.2) and (2.3). These paths have correct statistical distributions

and hence can be used to estimate values such as expected value of an observable g at time T, i.e. $E[g(X(t))]$. We will discuss three exact methods: 1. *Stochastic Simulation Algorithm* (SSA), 2. *Next Reaction Method* (NRM) and 3. *Modified Next Reaction Method* (MNRM) **??**.

2.3.1 Stochastic Simulation Algorithm (SSA)

In **?**, SSA was popularised by Gillepsie to simulate paths of SRNs that could be used to describe evolution of chemical reactions. The algorithms involves sampling two random numbers at each step: i) first random number is used to sample the time to next reaction, lets say h, and ii) second random number is used to sample which reaction takes place after time h. The algorithm is based on an observation from (2.2) that, given $X(t) = x$ at time t, the probability of $\mathcal{R}_j$ being the only reaction that takes place in time interval $(t, t+h)$ is given by $a_j(x) \times exp(-a_0(x)h)$ which can be split and rewritten as $(a_j(x)/a_0(x)) \times a_0(x)exp(-a_0(x)h)$.

As observed from (2.4), given $X(t) = x$ at time t, time to next reaction has an exponential distribution with parameter $a_0(x)$. Hence, the probability of firing reaction $\mathcal{R}_j$ in time interval $(t, t+h)$ can be split into two independent probabilities: probability of selecting reaction $\mathcal{R}_j$ from set of all the reactions, and probability density that follows an exponential distribution with parameter $a_0(x)$.

SSA involves the following sequence of steps to produce an exact simulation of a path of an SRN $X(t), t \geq 0$:

Algorithm 1 Stochastic Simulation Method

Algorithm: Stochastic Simulation Method
Initialize $\mathbf{x} = x_0$, $\mathbf{t} = 0$
Declare array $\mathbf{a}$
Declare constant $\mathbf{T}$, array ν
while $\mathbf{t} < \mathbf{T}$ **do**
  for each reaction-channel **i do**
    Calculate propensity $\mathbf{a}[\mathbf{i}]$ using x
  end for
  Set $\mathbf{a_0} = \sum_j \mathbf{a}[\mathbf{j}]$
  Generate a random number $\mathbf{r_1}$ from $unif(0,1)$
  Set $\mathbf{h} = \frac{1}{\mathbf{a_0}} \ln\left(\frac{1}{\mathbf{r_1}}\right)$
  Generate a random number $\mathbf{r_2}$ from $unif(0,1)$
  Find $\mathbf{m}$ such that $\sum_{k=1}^{m-1} \mathbf{a}[\mathbf{k}] < \mathbf{a_0 r_2} \leq \sum_{k=1}^{m} \mathbf{a}[\mathbf{k}]$
  Set $\mathbf{x} = \mathbf{x} + \nu[\mathbf{m}]$, $\mathbf{t} = \mathbf{t} + \mathbf{h}$
end while

It can be observed from the algorithm that for each reaction that is fired in the simulation, two random numbers are being generated, first to sample the time to next reaction and second to sample the reaction that should be fired based on their propensities. We will see in the following sections that NRM and MNRM require generation of only one random number each time a reaction is fired.

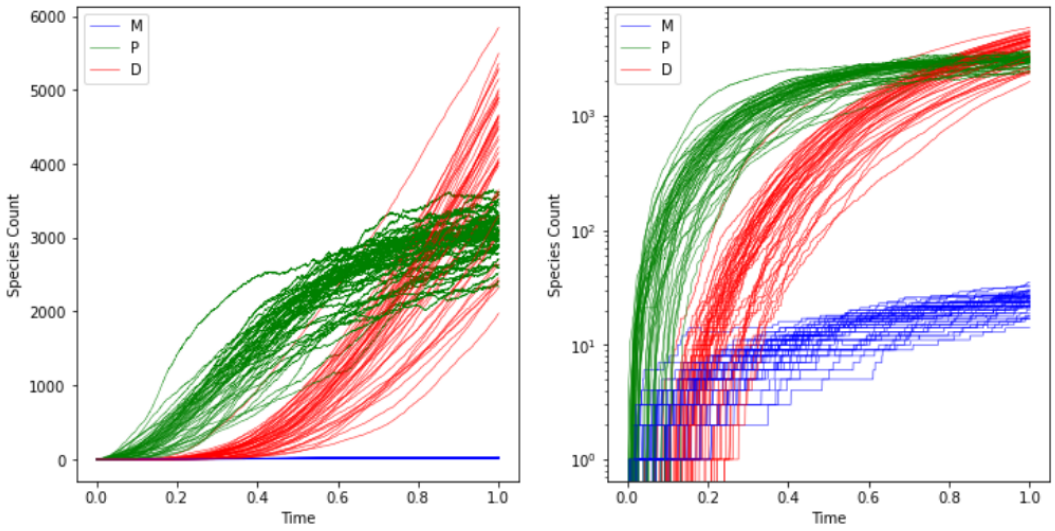


Figure 2.1: Fifty paths of gene transcription and translation SRN model simulated using SSA, with the left image showing a linear scale y-axis and the right image displaying a log scale y-axis.

The plots in Figure 2.1 show 50 exact paths of the gene transcription and translation SRN discussed above using Stochastic Simulation Algorithm.

2.3.2 Next Reaction Method (NRM)

The representation (2.7) of SRN paths as shown by Kurtz is given as, assuming SRN $\mathbf{X}$ is initially at state x_0:

$$X(t) = x_0 + \sum_{j=1}^{m} \nu_j Y_j \left(\int_0^t a_i(X(s))ds \right).$$

The representation (2.7) separates the randomness of the system from the state of the system, as the randomness is only contained in the Y_js which are unit-rate independent Poisson processes that count the number of times the corresponding reactions $\mathcal{R}_j$ fired until time t. The intensities of the Y_js are given by $a_j(X(t))$. Hence, each Y_j comes with its own time-frame, apart from the actual time frame of the chemical system. We can define $T_j(t) = \int_0^t a_j(X(s))ds$, to represent the *internal times* of the corresponding Y_j. Using internal times $T_j(t)$ in simulating SRN paths requires solving two crucial challenges: i) how to sample firing times of each Y_j in its own time frame and ii) how to translate these firing times to the actual time of the chemical system. We will see that NRM and MNRM try to solve these challenges in two different ways but both essentially use the same property of separation of randomness from the state of the system as suggested in (2.7).

For an SRN with M reactions, NRM first calculates the firing times of all the reactions, say $\delta t_j, j \leq M$ at time $t = 0$ by sampling from exponential distributions with internal times of Y_js as parameters, based on the assumption the a_j remain unchanged until j^th reaction is fired.

The reaction with the least firing time is fired first. After firing a reaction, the change in state of the system leads to change in propensities of the reaction-channels. This change in propensities requires the firing times to be updated, but the internal times of Y_j until their next firing remain the same. In the subsequent steps, the reaction with the nearest firing time is fired, and the firing times of the rest of the reactions are updated accordingly. For the reaction that got fired, new reaction time is sampled based on the updated propensity. The simulation is carried out until it is explicitly stopped or time limit of the simulation is reached.

Algorithm 2 Next Reaction Method

Algorithm: Next Reaction Method
Initialize $\mathbf{x} \leftarrow x_0$, $\mathbf{t} \leftarrow 0$
Declare array $\mathbf{a}$, array τ
Declare constant $\mathbf{T}$, array ν
for each reaction-channel **i** **do**
  Calculate propensity $\mathbf{a}[\mathbf{i}]$ using x
  Generate random number $\mathbf{r}$ from $unif(0,1)$
  Set $\tau[\mathbf{i}] = (1/a[i])\ln(1/r)$
end for
while $\mathbf{t} < \mathbf{T}$ **do**
  Set $\mathbf{t_{next}} = \min_j \tau[\mathbf{j}]$, $\mu = \arg\min_j \tau[\mathbf{j}]$
  Set $\mathbf{x} = \mathbf{x} + \nu[\mu]$, $\mathbf{t} = \mathbf{t_{next}}$
  for each reaction-channel **i** **do**
    Calculate propensity $\overline{\mathbf{a}}[\mathbf{i}]$ using x
    if $\mathbf{i} \neq \mu$ **then**
      Set $\tau[\mathbf{i}] = \mathbf{t} + (\mathbf{a}[\mathbf{i}]/\overline{\mathbf{a}}[\mathbf{i}])(\tau[\mathbf{i}] - \mathbf{t})$
    end if
  end for
  Generate a random number $\mathbf{r}$ from $unif(0,1)$
  Set $\tau[\mu] = \mathbf{t} + (\mathbf{1}/\overline{\mathbf{a}}[\mu])\ln(\mathbf{1}/\mathbf{r})$
  for each reaction-channel **i** **do**
    Set $\mathbf{a}[\mathbf{i}] = \overline{a}[i]$
  end for
end while

Lets say, at some time t, we know $X(t)$, internal times $T_j = T_j(t)$ and propensities $a_j = a_j(X(t))$. Assuming a_j remains constant over the next firing time of Y_j, we can also derive δt_j for each reaction. Now, the next reaction after time t that has to be fired should be the one with the minimum δt_j. Let us denote this value, i.e. $min_j \delta t_j$ as Δ, and let μ be the value of j with minimum δt_j. The system can then be updated to $\bar{t} = t + \Delta$ and all the propensities can be updated to $\overline{a}_j = a_j(X(\bar{t}))$. For $j = \mu$, next firing time should be sampled using the updated propensity $a_\mu(X(\bar{t}))$. As the propensity functions have changed for all the reactions, the next firing time would not be same as old firing times for $j \neq \mu$. But it is to be noted that the internal times of the reactions until their next firing remain the same. Internal times that have passed until $\bar{t}$ are given by $T_j(\bar{t}) = T_j(t) + a_j\Delta$. Hence the amount of internal time that should pass

before firing Y_j is given by

$$(T_j(t) + a_j \delta t_j) - (T_j(t) + a_j \Delta) = a_j(\delta t_j - \Delta).$$

Also, we note that the difference in internal times between t and $\bar{t}$ for Y_j are given by $\overline{a}_j \delta \bar{t}_j$. Hence, equating the differences between internal times for reactions, we get

$$\delta \bar{t}_j = \frac{a_j}{\overline{a}_j}(\delta t_j - \Delta).$$

NRM generates M random numbers during initialisation to sample the firing times of all the reactions. After the first step, it requires only one random variable per firing to sample the next firing time of the reaction that fired in the current step. NRM algorithm involves sequence of steps as shown in Algorithm 2.

Unlike SSA where two random variables were required in each step, NRM requires only one random number per firing of a reaction.

2.3.3 Modified Next Reaction Method (MNRM)

Algorithm 3 Modified Next Reaction Method

Algorithm: Modified Next Reaction Method
Initialize $\mathbf{x} = x_0$, $\mathbf{t} = 0$, array $\mathbf{P} = 0$, array $\mathbf{T} = 0$
Declare array $\mathbf{a}$, array δt
Declare constant $\mathbf{T}$, array ν
for each reaction-channel **i do**
  Calculate propensity $\mathbf{a}[\mathbf{i}]$ using $\mathbf{x}$
  Generate random number $\mathbf{r}$ from $unif(0, 1)$
  Set $\mathbf{P}[\mathbf{i}] = (1/a[i]) \ln(1/r)$
end for
while $\mathbf{t} < \mathbf{T}$ **do**
  for each reaction-channel **i do**
    Set $\delta t[i] = (P[i] - T[i])/a[i]$
  end for
  Set $\Delta = \min_i \delta t[i]$, $\mu = \arg\min_i \delta t[i]$
  Set $\mathbf{t} = \mathbf{t} + \mathbf{\Delta}$, $\mathbf{x} = \mathbf{x} + \nu[\mu]$
  for each reaction-channel **i do**
    Set $\mathbf{T}[\mathbf{i}] = \mathbf{T}[\mathbf{i}] + \mathbf{a}[\mathbf{i}]\mathbf{\Delta}$
  end for
  Generate a random number $\mathbf{r}$ from $uniform(0, 1)$
  Set $\mathbf{P}[\mu] = \mathbf{P}[\mu] + \ln(\mathbf{1}/\mathbf{r})$
  for each reaction-channel **i do**
    Calculate $\mathbf{a}[\mathbf{i}]$ using $\mathbf{x}$
  end for
end while

MNRM is similar to NRM in the way that it uses Kurtz' representation of evolution of X to

simulate paths of $\mathbf{X}$, but unlike NRM, it does not require time conversions and works explicitly with internal times to calculate next firing times of reactions. At some time t, let P_j denote the first firing internal time of Y_j that is strictly larger than T_j: $P_j = min_s\{s > T_j : Y_j(s) > Y_j(T_j)\}$, where T_js are the internal times of Y_js until t. Hence the next firing time of the reaction can be given as:

$$\delta t_j = (P_j - T_j)/a_j.$$

The internal time until the next firing of a reaction Y_j should remain constant irrespective of the evolution of the system until Y_j is fired, only the actual firing time changes based on the changes in propensities. MNRM uses this fact and lets us sample internal times, here P_j, directly instead of sampling actual next times of reactions. Algorithm 3 shows the sequence of steps involved in MNRM.

MNRM calculates M random numbers during initialisation to sample the internal times of first firings of all the Y_js, and subsequently uses only one random number per iteration to re-sample the internal time of the fired reaction. MNRM and NRM differ in the way they use random numbers. MNRM utilizes random numbers to sample the internal times until the first firing, eliminating the need for time-conversions as in NRM.

2.3.4 Approximate Algorithms - Tau-Leap Algorithm

Exact algorithms, although being useful in simulating paths of SRNs with correct distributions, may not be feasible for many reasons. High propensities can lead to lower and lower reaction times, and hence increase computation load of the algorithms. Approximate algorithms mitigate these challenges by simulating paths of SRNs over discrete time steps. One such algorithm is the Tau-leap algorithm **?**.

In this algorithm, a small time interval τ is fixed and a path of an SRN $\mathbf{X}$ is simulated by calculating $X(t)$ at $t = 0, \tau, 2\tau, \ldots, N\tau$ where $N\tau$ is the time limit of the simulation. After k time-steps, when $t = k\tau$, the algorithm progresses by sampling the number of each kind of reaction that would take place in the interval $(k\tau, (k+1)\tau]$ and update the state of the system accordingly. Let the number of reactions that take place in the interval $(k\tau, (k+1)\tau]$ for M reaction channels be $(\mu_{k+1,1}, \mu_{k+1,2}, \ldots, \mu_{k+1,M})$, then the system is updated as follows:

$$X((k+1)\tau) = X(k\tau) + \sum_{j=1}^{M} \mu_{k+1,j}\nu_j.$$

If the initial state of the SRN is $X(0) = x_0$, and if the time interval $(0, T]$ is partitioned into N time intervals, $(0, s_1], (s_1, s_2], \ldots, (s_{N-1}, s_N]$, the integral $\int_0^t a_j(X(s))ds$ in Kurtz's representation (2.7) can be approximated as a summation over discrete time intervals so that we get

$$\int_0^t a_j(X(s))ds \approx \sum_{q=0}^{N-1} a_j(X(s_q))(s_{q+1} - s_q).$$

Let us represent the path simulated by approximate methods by $\overline{X}(t)$ to differentiate it from exact paths. The approximate path $\overline{X}$ that can be modeled by approximating the integrals in random time change representation (2.7) is thus given as:

$$\overline{X}(t) = x_0 + \sum_{j=1}^{m} Y_j \left(\sum_{q=0}^{N-1} a_j(\overline{X}(s_q))(s_{q+1} - s_q) \right) \nu_j. \tag{2.8}$$

While simulating $\overline{X}(t+\tau)$ given the state $\overline{X}(t)$, the second term in (2.8) becomes

$$Y_j \left(\sum_{q=0}^{N-1} a_j(\overline{X}(s_q))(s_{q+1} - s_q) + a_j(\overline{X}(t))\tau \right).$$

Y_j becomes a Poisson process with internal time as sum of two terms. Hence, Y_j can be split into two terms: $Y_j \left(\sum_{q=0}^{N-1} a_j(\overline{X}(s_q))(s_{q+1} - s_q) \right)$ and a Poisson random Variable $\mathcal{P}_j(a_j(\overline{X}(t))\tau)$. The first term is already known as it was used to arrive at the state $\overline{X}(t)$. Hence the expression for $\overline{X}(t+\tau)$ becomes:

$$\overline{X}(t+\tau) = \overline{X}(t) + \sum_{j=1}^{M} \mathcal{P}_j(a_j(x)\tau)\nu_j,$$

where $\mathcal{P}_j(\lambda_j), j = 1, 2, \ldots, M$, are independent Poisson processes with intensity λ_j.

Tau-Leap algorithm involves the following sequence of steps:

Algorithm 4 Tau-Leap Algorithm

Initialize $\mathbf{x} = x_0$, $\mathbf{t} = 0$
Declare array $\mathbf{a}$, array $\mathbf{p}$
Declare constant τ_0, $\mathbf{T}$, array ν
while $\mathbf{t} < \mathbf{T}$ **do**
  for each reaction-channel **i do**
    Calculate propensity $\mathbf{a}[\mathbf{i}]$ using $\mathbf{x}$
    Generate random number $\mathbf{p}[\mathbf{i}]$ from Poisson distribution $\mathcal{P}(\mathbf{a}[\mathbf{i}]\tau)$
  end for
  Set $\mathbf{t} = \mathbf{t} + \tau, \mathbf{x} = \mathbf{x} + \sum_{\mathbf{j}=\mathbf{1}}^{\mathbf{M}} \mathbf{p}[\mathbf{j}]\nu[\mathbf{j}]$
end while

The plots in Figure 2.2 show 50 approximate paths of gene transcription and translation SRN simulated using tau-leap algorithm.

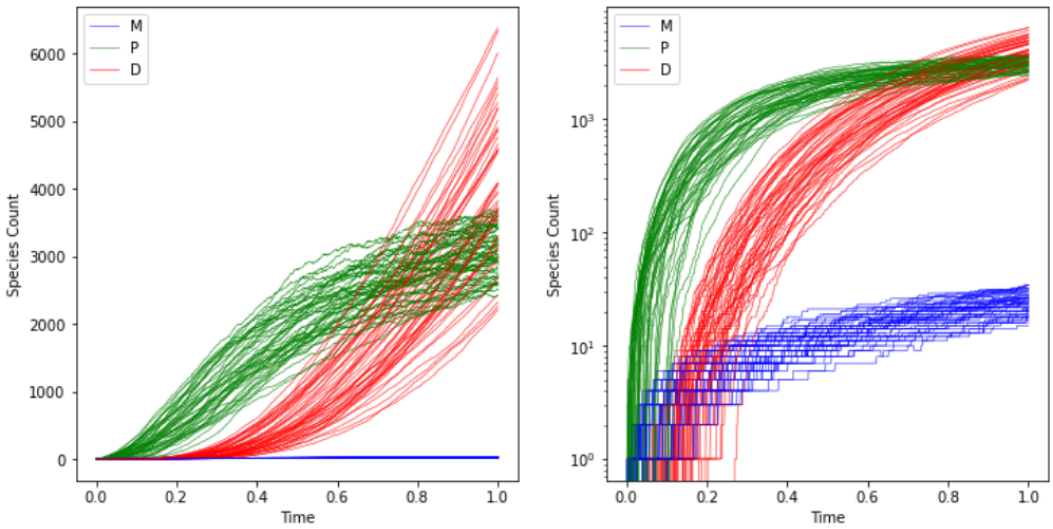


Figure 2.2: Fifty approximate paths of gene transcription and translation SRN model simulated using tau-leap method, with the left image showing a linear scale y-axis and the right image showing a log scale y-axis.

Chapter 3

Analogy between Supply Chains and Chemical Reaction Modelling

As seen in chapter 2 - chemical reaction modelling, simulations can be useful in modelling the dynamic behaviour of systems with multiple moving parts (molecules) and processes (reactions). Supply chains involve coordinated transportation of materials from suppliers to consumers. As global supply chains grow more and more complex, simulations play an important role in designing of policies prior to implementation by performing sensitivity-analysis, and ensuring resiliency and sustainability by being able to incorporating several scenarios in the simulations.

One of the most widely used algorithms to exactly simulate supply chains is the discrete-event simulation (DES) algorithm, that involves instantaneous change in state of the system at distinct points in time **?**. Due to computational and performance related shortcomings of exact methods such as DES, approximate methods can be employed for simulations such as the time-bucket method, that involves simulating paths of the supply chain at discrete time intervals. These methods accelerate the simulations, as system state is updated only at fixed time-intervals, unlike exact methods where each distinct event leads to a change in state of the system.

In this chapter, we discuss DES, time-bucket method and L-Leap method, that is an extension of D-Leap method **?**, and explore the similarities between supply chain modelling and chemical reaction modelling using SRNs. We will incorporate delays in the processes in supply chain simulations, following the instantaneous-consumption-delayed-production approach **?**.

3.1 DES of supply chains

The DES **?** is a widely used algorithm to simulate logistics and supply chains. The *events* and *states* in DES can be modelled in accordance to the specific goal and use-case of the simulation. Each event taking place in a simulation results in an instantaneous change in the state of the system. One kind of DES is the next-event time-advance algorithm where the time leaps to the

next nearest event among the list of future events. It is worth noting that next-event time-advance is analogous to SSA, that was mentioned in section (2.3.1). Events in the DES algorithm are analogous to the chemical reactions in SSA, and the states are analogous to the number of molecules in the chemical system being simulated by SSA. Similar to SSA and other exact methods of simulation SRNs, the complexity of DES depends on the number of events that take place within a time period. Using DES for simulating complex supply chains with high number of processes and high production rates can be computationally expensive. Approximate methods, such as the time-bucket method, accelerate simulations by calculating state after fixed time intervals. Approximate methods circumvent the computational challenges posed by exact methods by changing state of the system at fixed time intervals, and not separately for each discrete event.

3.2 Time-bucket method

In the time-bucket method **?**, a time interval τ is fixed and the algorithm progresses by simulating the number of each event that takes place in the time intervals $(0, \tau], (\tau, 2\tau], \ldots$ until the time limit of the simulation is reached, unlike the DES where each discrete event is simulated at their exact simulated times. The time-bucket method is, thus, an approximation of the DES, similar to how the tau-leap algorithm (2.3.4) used in simulating evolution of chemical systems, is an approximation of the exact method SSA. Hence, time-bucket method is often useful in simulating complex supply chains with large number of events and states.

In the time-bucket method simulation of a supply chain, each event represents a process in the supply chain, and we consider $(P_1, P_2, \ldots, P_d)$ as the supply chain's parts that are manufactured and transported between different processes. These parts are analogous to the species of molecules $(S_1, S_2, \ldots, S_d)$ in a chemical reaction as discussed in section **??**. Given J processes, each process $j \in \{j \mid j \in \mathbb{Z}, 1 \leq j \leq J\}$ can be represented as follows:

$$\sum_{i \in A_j} c_{ij} P_i \rightarrow \sum_{i \in B_j} p_{ij} P_i,$$

where A_j and B_j are the sets of parts consumed and produced respectively by process j, and c_{ij} and p_{ij} are the numbers of part P_i consumed and produced respectively by process j. If we consider c_{ij} and p_{ij} as the i^{th} elements of vectors $C_j \in \mathbb{Z}_+^d$ and $P_j \in \mathbb{Z}_+^d$ respectively, the vector $\nu_j = P_j - C_j$ is equivalent to the stoichiometric vector ν_j of a reaction (here process) j. Given state $X(t) = (x_1(t), \ldots, x_d(t))$ at time t, each process $j \in \{j \mid j \in \mathbb{Z}, 1 \leq j \leq J\}$ is also associated with a production rate $\lambda_j(X(t))$ that describes the rate of consumption and production of parts. The maximum value that $\lambda_j(X(t))$ can take is a fixed value represented by λ_j^{max}. The rate of production by process j at time t depends on the availability of all parts $P \in A_j$ at time t as well as possible disruptions during the time period of simulation. Less availability of the parts hampers the rate of the processes, leading to less number of occurrences

of the processes in the time intervals.

In a single simulation of a supply chain using the time-bucket method, a time interval τ is fixed and the approximate state of the system $\overline{X}(t) \in \mathbb{Z}_+^d$ is simulated at $t = \tau, 2\tau, \ldots, N\tau$ where $T = N\tau$ is the time limit of the simulation. The state $\overline{X}(t+\tau)$, given the state $\overline{X}(t)$ at time t is updated as follows:

$$\overline{X}(t+\tau) = \overline{X}(t) + \sum_{j=1}^{J} \mathcal{P}_j(\lambda_j(\overline{X}(t))\tau)\nu_j, \tag{3.1}$$

where $\mathcal{P}_i$s are independent Poisson processes with rates given by $\lambda_j(\overline{X}(t))\tau$. At each time step, each $\overline{x}_i(t+\tau)$ in $\overline{X}(t+\tau) = (\overline{x}_1(t+\tau), \overline{x}_2(t+\tau), \ldots, \overline{x}_d(t+\tau))$ is checked for negativity, and the number of processes simulated in the time period $(t, t+\tau]$ are adjusted accordingly.

Let us assume that $\overline{X}^{temp}(t+\tau) = (\overline{x}_1^{temp}(t+\tau), \ldots, \overline{x}_d^{temp}(t+\tau))$ is the updated state obtained after (3.1) in the time interval $(t, t+\tau]$, and let $\overline{x}_i^{temp}(t+\tau)$ be the least negative number in $\overline{X}^{temp}(t+\tau)$. For the process j that consumes c_{ij} number of parts P_i in a single occurrence, the number of simulated occurrences of process j in the interval $(t, t+\tau]$ is reduced by -$\lfloor \overline{x}_i^{temp}(t+\tau)/c_{ij} \rfloor$. Hence the number of parts of P_i at time $t+\tau$ is updated as follows:

$$\overline{x}_i^{temp}(t+\tau) = \overline{x}_i^{temp}(t+\tau) - c_{ij}\lfloor \overline{x}_i^{temp}(t+\tau)/c_{ij} \rfloor.$$

It should be noted that $-c_{ij}\lfloor \overline{x}_i^{temp}(t+\tau)/c_{ij} \rfloor$ is a positive number that, when added to $\overline{x}_i^{temp}(t+\tau)$, makes $\overline{x}_i^{temp}(t+\tau)$ a non-negative number. This process is repeated until all the negative numbers in state $\overline{X}^{temp}(t+\tau)$ have been eliminated, and finally $\overline{X}(t+\tau)$ is set equal to $\overline{X}^{temp}(t+\tau)$. $\overline{X}(t+\tau)$ can then be used in the subsequent time step in the time-bucket algorithm.

3.3 Logistic-Leap (L-Leap) Method

Each process in a supply chain is usually associated with a delay. A delay for a process j is the amount of time that passes between consumption and production of parts by that process. L-Leap method **?** extends D-Leap method **?** used in simulating chemical reactions with delays, by incorporating several features of a supply chain such as transportation, production time, inventory management and push and pull systems. The simulation of supply chains using time-bucket method can be split into two phases: instantaneous consumption and delayed production. This is based on the idea that each process in a supply chain consumes parts instantaneously and takes some finite non-zero time period to produce parts.

3.3.1 Consumption

Each occurrence of any process j is assumed to consume parts instantaneously and produce parts with some delay. Let $C_j = (c_{j1}, c_{j1}, \ldots, c_{jd})$ represent the vector containing number of

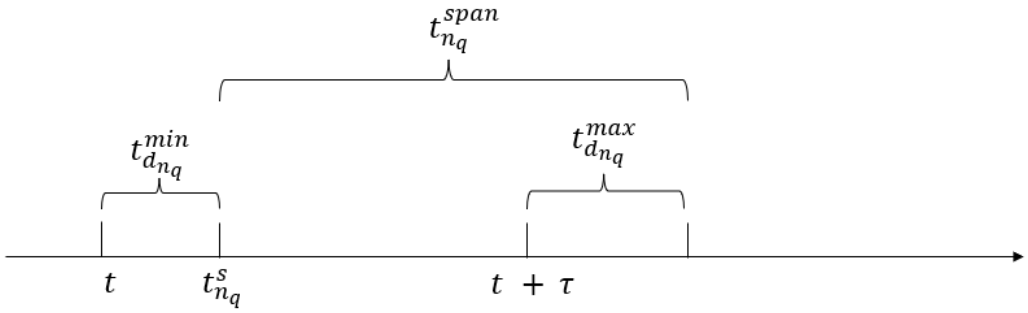


Figure 3.1: Timeline for processes simulated during time interval $(t, t + \tau]$.

parts consumed during a single occurrence of a process j. During a simulation, given state $\overline{X}(t)$ at time t, consumption is simulated by calculating the number of each process j in time interval $(t, t + \tau]$ and updating the state as follows:

$$\overline{X}(t + \tau) = \overline{X}(t) - \sum_{j=1}^{J} \mathcal{P}_j(\lambda_j(\overline{X}(t))\tau)C_j. \tag{3.2}$$

Each element in $\overline{X}(t + \tau)$ is checked for negativity, and the number of processes simulated in $(t, t + \tau]$ are updated to ensure non-negativity of $\overline{X}(t + \tau)$. The $\mathcal{P}_j$s are independent Poisson processes with rates $\lambda_j(\overline{X}(t))\tau$ respectively.

Each process $j \leq J$ can be associated with a range $[t_j^{min}, t_j^{max}]$ that denotes the range of time delays the process j can have between consumption and production. With time delays in processes, the state of the system is influenced not only by its immediate predecessor but also by delayed processes, rendering the process non-Markovian. In order to make the system Markovian, the algorithm in **?** extends the state by incorporating each delayed process simulated in the time interval $(t, t + \tau]$ into the state of the system. This is required in order to update the state of the system appropriately when these processes produce their respective parts after delays. To achieve this, in **?**, a queue structure is maintained where each entry stores the following information, assuming q_j occurrences of process j are simulated in time interval $(t, t + \tau]$:

- The index of delayed process $d_{n_q} = j$.
- Number of delayed processes $Q_{n_q} = q_j$.
- Earliest time of completion of delayed process d_{n_q} represented by $t_{n_q}^s = t + t_j^{min}$.
- Time span of completion of Q_{n_q} processes represented by $t_{n_q}^{span} = t + \tau + t_j^{max} - (t + t_j^{min}) = \tau + t_j^{max} - t_j^{min}$.

3.3.2 Production

After adjusting the state of the system by incorporating consumption by processes simulated in the interval $(t, t+\tau]$, the algorithm iterates through to queue to determine the number of processes that completed in the current time bucket. For an entry n_q, if $t^s_{n_q} > t+\tau$, i.e. earliest time of completion of process d_{n_q} does not lie in the interval $(t, t+\tau]$, then the entry is skipped. Similarly, an entry n_q is skipped if $Q_{n_q} = 0$. For entries not lying in the above criteria, a fraction of the Q_{n_q} processes will be completed in the current time bucket. Let $P_j = (p_{j1}, p_{j1}, \ldots, p_{jd})$ represent the vector containing number of parts produced during a single occurrence of a process j. The following updates are made to the queue entries and the state of the system:

- Number of completed processes K_{n_q} are modeled by a Binomial distribution with parameters Q_{n_q} and $min\left(\frac{t+\tau-t^s_{n_q}}{t^{span}_{n_q}}, 1\right)$.
- Number of delayed processes is updated: $Q_{n_q} = Q_{n_q} - K_{n_q}$.
- Earliest time of completion is updated: $t^s_{n_q} = t+\tau$.
- Span is updated: $t^{span}_{n_q} = max(0, t^{span}_{n_q} - (t+\tau-t^s_{n_q}))$.
- Finally, the state of system is updated as follows:

$$\overline{X}(t+\tau) = \overline{X}(t) + K_{n_q} P_{n_q}.$$

3.4 Monte Carlo methods in supply chain management

Monte Carlo methods **?** are computational algorithms that rely on random sampling to analyse complex systems and obtain numerical results, that are otherwise difficult to obtain analytically. These are particularly useful in systems that have input parameters as random variables, following some probability distributions. Monte Carlo methods are used to simulate the paths of systems by repeatedly sampling from probability distributions, and the quantities of interest are calculated by aggregating over the simulated paths. As the method depends on randomness to produce numerical results for the quantities of interest, the two most prominent errors that are considered while simulating a system using MC methods are:

- Sampling error - Error that arises due to the inherent random nature of the MC method. As parameters are repeatedly sampled from probability distributions while simulating paths of a system, each path can give different numerical result for the quantity of interest, and only the aggregated value over a number of simulations can give a reliable value. This error can be minimised by aggregating over a large number of simulated paths.
- Bias error - Error that occurs due to the approximations incorporated in the simulations. This error usually arises when the simulation algorithm does not represent a real-world

system. In case of time-bucket method, the length of time-interval affects the bias error. We recall that time-bucket method and tau-leap method are approximations of exact methods such as DES and SSA.

MLMC (Multilevel Monte Carlo) **?** methods offer a significant improvement over traditional Monte Carlo techniques in terms of computational cost by efficiently estimating quantities of interest using a hierarchy of discretisations. It enables maintaining the appropriate level of resolution by combining multiple levels of resolutions in a hierarchy.

If the aim of simulations is to carry out sensitivity analysis based on input parameters, then another layer of simulations have to be performed to assess the effect of different input parameters on the output quantities. This approach is known as Multi Level Monte Carlo (MLMC) method **?**.

While simulating supply chains, the system is usually simulated a large number of times to minimise the sampling error and to obtain a reliable value for the quantity of interest. We see an example of using MC method to simulate a supply chain with three supply parts and five processes below.

Example - Simulating a supply chain with three supply parts and five processes

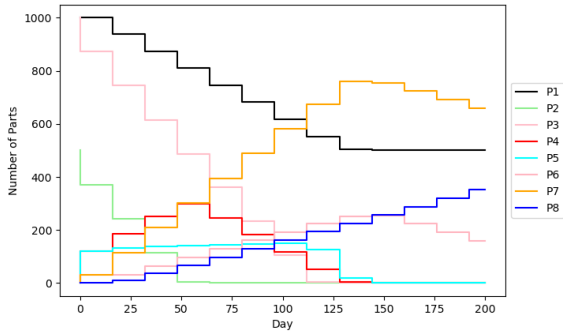
We use MC method to simulate the supply chain represented in Figure 1.1 with eight parts $P_1, P_2, \ldots, P_8$, and five processes given as:

$$Process1 : P_2 \rightarrow P_4,\ Process2 : P_3 \rightarrow P_5,\ Process3 : P_1 + P_4 \rightarrow P_6,$$

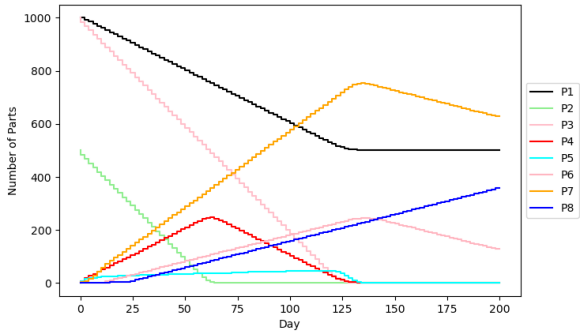
$$Process4 : P_5 \rightarrow P_7\ and\ Process5 : P_6 + P_7 \rightarrow P_8.$$

The initial state is assumed to be: $x_1(t = 0) = 1000,\ x_2(t = 0) = 500,\ x_3(t = 0) = 1000,\ x_i(t = 0) = 0\ \forall\ i\ \in \{4, 5, 6, 7, 8\}$. P_1, P_2 and P_3 are the initial supplies of the supply chain, that are consumed by processes $3, 1$ and 2 respectively. The production rates of the processes are assumed to be $\lambda_1 = 8, \lambda_2 = 8, \lambda_3 = 4, \lambda_4 = 8$ and $\lambda_5 = 2$. It is also assumed that the time delays of production by the processes are deterministic, i.e. $t_j^{min} = t_j^{max}$ for $j \in \{j \mid 1 \leq j \leq 5\}$. In this example, the time delays are assumed to be $t_1^{min} = 1, t_2^{min} = 1, t_3^{min} = 10, t_4^{min} = 1$ and $t_5^{min} = 10$.

It can be seen from Figure 3.2 that both time-buckets, i.e. $\tau = 2$ and $\tau = 16$ are able to capture dynamics of the supply chain. Process 3 consumes parts P_1 and P_4 to produce P_6, and it is evident from the plots that the consumption of P_1 stops at $x_1 = 500$ units, which is right after entire P_4 parts are consumed. Moreover, x_8 increases linearly after some waiting period, which is attributed to the production delays of the processes that are involved in production of parts of P_8.



(a) Time history with $\tau = 16$ days.



(b) Time history with $\tau = 2$ days.

Figure 3.2: Time histories of the supply chain represented in Figure 1.1, 200 MC simulations.

3.5 Inventory Management

Inventory management plays an important role in a push system. Maintaining a larger inventory carries the burden of higher storage and maintenance costs while also increasing the risk of items becoming obsolete over time. As highlighted in the study by Paschalidis **?**, the challenge in inventory control lies in striking a balance between the costs associated with *producing* more goods, which raises inventory and management expenses, and *idling*, which may result in stock-outs and unmet customer demands. We look at the behaviour of the supply chain simulated in the last section under varying initial amounts of supply parts P_1, P_2 and P_3 over long simulation times.

In Figure 3.3, we simulate the supply chain mentioned in the previous section with the initial state: $x_1(t = 0) = 1000,\ x_2(t = 0) = 500,\ x_3(t = 0) = 1000,\ x_i(t = 0) = 0\ \forall\ i \in \{4, 5, 6, 7, 8\}$ over a period of 400 days. Rates and production delays of the processes are assumed to be the same as the last section. From the plot, we can see a case of excessive supply of part P_1. The consumption of P_1 ceases right after $Process\ 3$ stops occurring due to the shortage of part P_4. The early shortage of P_4 can be attributed to the insufficient initial supply of part P_2, which was lower than the initial supply of part P_1. As a result, the limited availability of P_2 hindered the production of P_4, leading to the observed shortage during the early stages.

Moreover, we also observe that the consumption of part P_7 stops right after entire P_6 gets consumed, as $Process\ 5$ could not occur to produce part P_8. The early consumption of part P_6 can be attributed to the early consumption of P_4, which again is a consequence of insufficient supply of part P_2 at time $t = 0$.

To counter the early consumption of parts P_2, P_4 and P_6, and to reduce the surplus amount of part P_1 and P_7, we simulate the supply chain with the initial state: $x_1(t = 0) = 700,\ x_2(t = 0) = 500,\ x_3(t = 0) = 700,\ x_i(t = 0) = 0\ \forall\ i \in \{4, 5, 6, 7, 8\}$ in Figure 3.4.

We observe in Figure 3.4 that we get lesser surplus amounts of parts P_1 and P_7, i.e. $x_1(t =$

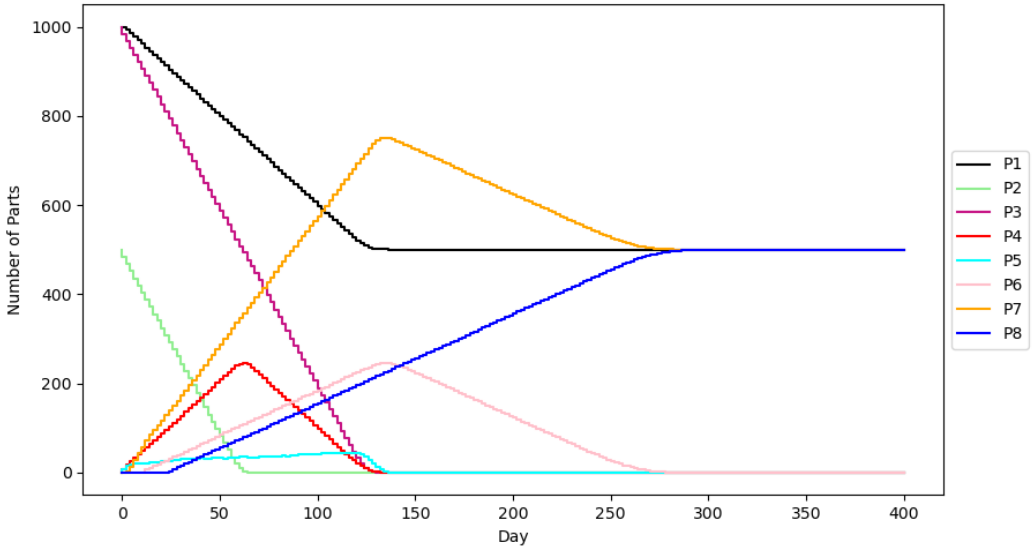


Figure 3.3: Time histories of the supply chain represented in Figure 1.1, aggregated over 100 simulations, $\tau = 2$ days, simulated over 400 days.

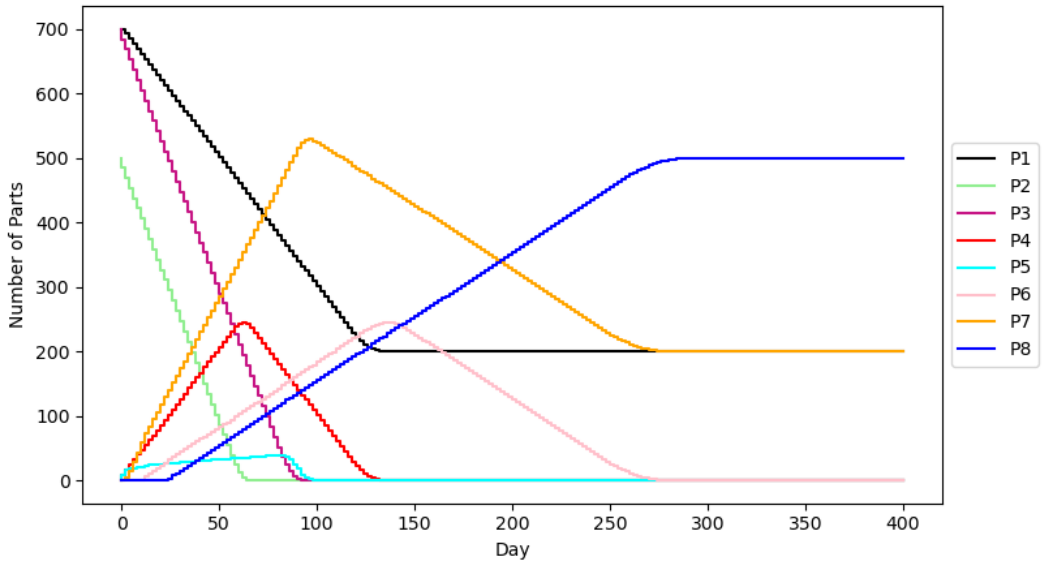
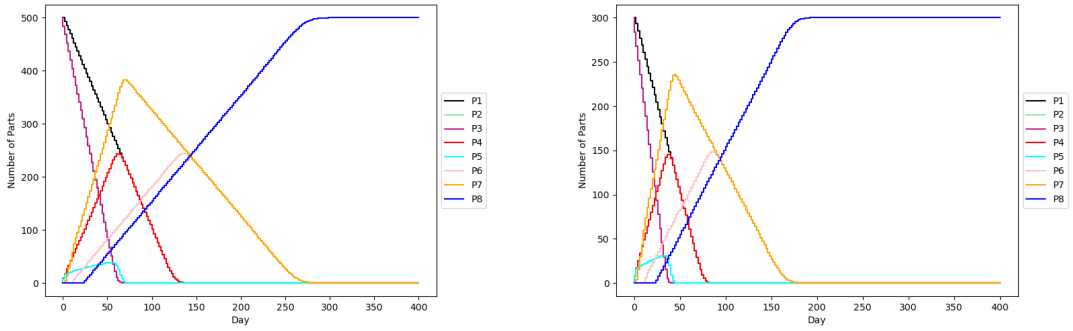


Figure 3.4: Time histories of the supply chain represented in Figure 1.1, aggregated over 100 simulations, $\tau = 2$ days, simulated over 400 days. Initial state:$x_1(t = 0) = 700,\ x_2(t = 0) = 500,\ x_3(t = 0) = 700,\ x_i(t = 0) = 0\ \forall\ i \in \{4, 5, 6, 7, 8\}$.

$400) = x_7(t = 400) = 200$, when compared to $x_1(t = 400) = x_7(t = 400) = 500$ for the previous initial state. In both the cases, the final amount of part P_8 is seen to be $x_8(t = 400) = 500$, which matches the initial amount of part P_2, i.e. $x_2(t = 0) = 500$. Hence, the initial amount of part P_2 acts as the limiting factor in production of the output part P_8, as it limits the consumption of parts P_1 and P_7, pointing towards the fact that the initial state should be adjusted such that there is no surplus inventory or surplus intermediate part at any point in period of supply chain. For this, we simulate the supply chain with the initial states with equal amounts of P_1, P_2 and P_3 in Figure 3.5.



(a) Time history with initial state: $x_1(t = 0) = 500,\ x_2(t = 0) = 500,\ x_3(t = 0) = 500$.

(b) Time history with initial state: $x_1(t = 0) = 300,\ x_2(t = 0) = 300,\ x_3(t = 0) = 300$.

Figure 3.5: Time histories of the supply chain represented in Figure 1.1, aggregated over 100 simulations, $\tau = 2$ days, simulated over 400 days.

In Figure 3.5, we finally observe that entire P_1, P_2 and P_3 get consumed, and no surplus of any intermediate part is seen in the period of simulation. We also simulate the supply chain with less inventory in Figure 3.5b, to observe the change in final number of output parts produced after the consumption of all the supply parts and intermediate parts. The number of output parts P_8 produced at $t = 400\ days$ is $x_8(t = 400) = 500$ in Figure 3.5a and $x_8(t = 400) = 300$ in Figure 3.5b.

Even after obtaining the optimal initial values of the supply parts in the supply chain depicted in Figure 1.1, we see that the production of output part P_8 ceases once all the inventory and intermediate parts are consumed by the supply chain processes. Supply chains require the implementation of effective strategies that guarantee uninterrupted production of products. This involves establishing a well-structured system for periodically stocking up on supply parts, driven by carefully defined criteria. One such strategy is the safety stock strategy.

3.5.1 Safety Stock Strategy

Safety stock is a popular strategy used to mitigate challenges posed by disruptions in supply chain operations, such as uncertainties in the supply, or variations in product demand. One of the methods is to fix a replenishment amount that has to be ordered every time the quantity of a

supply/raw material goes below a threshold value. Let $x_p^b(t)$ denote the replenishment order of part p at time t, then $x_p^b(t)$ is calculated as follows:

$$x_p^b(t) = \begin{cases} S_p, & \text{if } x_p(t) \leq x_p^s, \\ 0, & \text{otherwise,} \end{cases} \tag{3.3}$$

where x_p^s is known as the safety stock of supply p. If the stock of part p goes below the threshold, i.e. x_p^s, a replenishment order of S_p amount of part p is made to restock the supply part p. This order, often referred to as back order, usually has a time delay, let us denote it as t_p^b.

It is essential for a supply chain to strike a correct balance between S_p, x_p^s and t_p^b so that the intermediate parts along with the final product do not run out of stock, and the final product is produced at a predictable rate. It is also important to ensure that no intermediate part is getting accumulated because of excess of raw materials and fast intermediate processes. Large stock piles incur costs on maintenance and storage. Strategy to implement inventory control is usually problem specific, and some works can be found in **???**.

In the time-bucket simulation of supply chains, the following algorithm can be used to update the system based on replenishment orders, after updating the system based on consumption and production of parts. For a given simulation time T, a time stamp $M > T$, back order time delays $\{t_p^b\}$, safeguard orders $\{S_p\}$, set of supplies/raw materials $\mathbf{S}$ and safety stocks $\{x_p^s\}$, the safety stock method is given in algorithm 5.

Algorithm 5 Inventory Control - Safety Stock Algorithm

Declare constant $\bar{t}_p$, M, array S_p
while $t < T$ **do**
  Update system based on consumption and production algorithms
  for each $p \in \mathbf{S}$ **do**
    Compute the backorder quantity x_p^b using:
      if $x_p(t) \leq x_p^s$:
        $x_p^b = S_p$
      else:
        $x_p^b = 0$
    if $t \geq \bar{t}_p$ **then**
      Backorder arrived. Add it to the state vector: $x_p \leftarrow x_p + x_p^b$
      Reset the next arrival time: $\bar{t}_p \leftarrow M$
    else if $t < \bar{t}_p$ and $\bar{t}_p = M$ and $x_p^b > 0$ **then**
      Compute the next backorder arrival time: $\bar{t}_p \leftarrow t + t_p^b$
    end if
  end for
end while

In our example of supply chain, we implement the safety stock strategy with different set of values of safety stocks, safeguard orders and time-delays, with an aim to avoid stock-outs and surplus stocks.

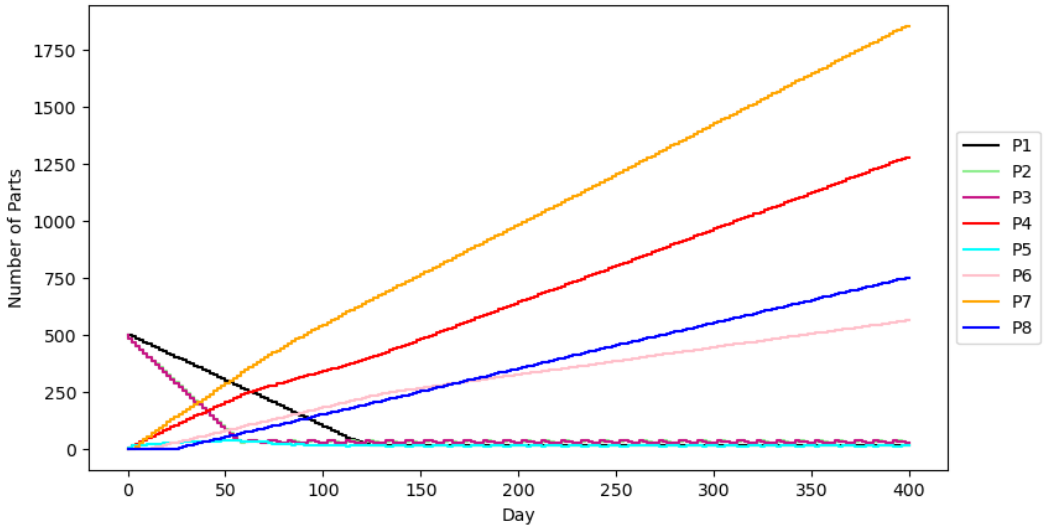


Figure 3.6: Time histories of the supply chain represented in Figure 1.1 after incorporating safety-stock strategy, aggregated over 100 simulations, $\tau = 2$ days, simulated over 400 days. Initial state: $\{500, 500, 500, 0, 0, 0, 0, 0\}$.

In the previous section, we assumed the production rates of our processes to be $\lambda_1 = 8, \lambda_2 = 8, \lambda_3 = 4, \lambda_4 = 8$ and $\lambda_5 = 2$. Our first approach of deciding safety stocks, safeguards orders and time-delays is to ensure uninterrupted operation of the processes that directly consume supply parts P_1, P_2 and P_3, i.e. $Process\ 3,\ 2$ and 1 respectively. Given the rates of these processes to be $\lambda_1 = 8, \lambda_2 = 8, \lambda_3 = 4$, and time-bucket $\tau = 2\ days$, the average occurrences of Processes $1, 2$ and 3 are $16, 16$ and 8 respectively in each time-bucket. Hence, in each time-bucket, 8 parts of P_1, 16 parts of P_2 and 16 parts of P_3 are consumed on average. Assuming a fixed back-order time of $t_1^b = t_2^b = t_3^b = 8\ days$ (i.e. 4 time-buckets), we can set the values of safety stock and safeguard orders such that there is enough amount of supply parts available for the processes $1, 2$ and 3 even after replenishment order is made and before the order arrives. Thus, we set the safety stocks to be:

$$\begin{aligned} x_1^s &= 5\ (4\ timebuckets\ for\ order\ arrival + 1\ timebucket\ for\ placing\ the\ order) \\ &\quad \times 8(average\ consumption\ per\ timebucket) \\ &= 40, \end{aligned}$$

$x_2^s = 5 \times 16 = 80$ and $x_3^s = 5 \times 16 = 80$, and safeguard orders to be $S_1 = 5 \times 8 = 40,\ S_2 = 5 \times 16 = 80$ and $S_1 = 5 \times 16 = 80$. Figure 3.6 shows the plot with the above values for the safety-stock parameters, and initial state: $\{500, 500, 500, 0, 0, 0, 0, 0\}$.

In Figure 3.6 we achieved a steady production of output part P_8 and the amounts of supply parts $P_1,\ P_2$ and P_3 remain steady over time, but there is significant accumulation of intermedi-

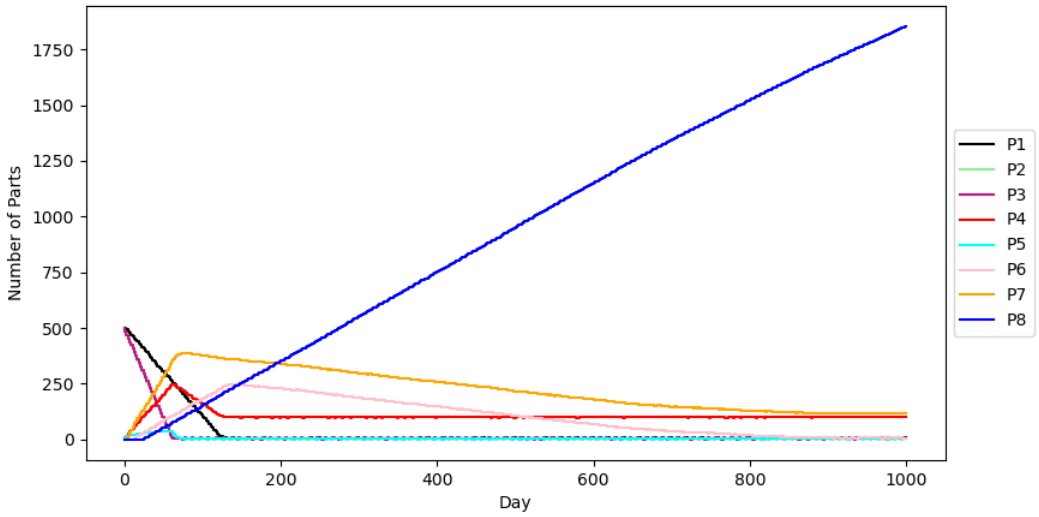


Figure 3.7: Time histories of the supply chain represented in Figure 1.1 after incorporating safety-stock strategy, aggregated over 100 simulations, $\tau = 2$ days, simulated over 1000 days. Initial state: $\{500, 500, 500, 0, 0, 0, 0, 0\}$. Safety stock for supply parts P_1, P_2 and P_3 is set to 16.

ate parts P_4, P_6 and P_7. This can be attributed to the lower rate of production of P_8 by Process 5 at $\lambda_5 = 2$. It is worth noting that the production rate of Process 5 is the lowest amongst all the intermediate processes. Hence, in order to achieve a steady production of output part P_8 without over-stocking intermediate parts, the safety-stocks and safeguard orders have to be adjusted such that the average production rate of all the intermediate processes stay close to 2. Considering time-bucket $\tau = 2\ days$, a production rate of 2 for each process would require an average consumption of 4 parts of each P_1, P_2 and P_3 in a time-bucket. Assuming the same back-order time of $t_1^b = t_2^b = t_3^b = 8\ days$ (i.e. 4 time-buckets), we set the safety-stocks and safeguard orders as follows: $x_1^s = 5\ (time\ buckets) \times 4\ (average\ consumption\ per\ timebucket) = 20$, $x_2^s = 5 \times 4 = 20$ and $x_3^s = 5 \times 4 = 20$, and assign safeguard-orders the same values as safety-stocks, i.e. $S_1 = 20$, $S_2 = 20$ and $S_3 = 20$. The supply chain is simulated over 1000 days and the aggregated time histories over 100 MC simulations is shown in Figure 3.7.

In Figure 3.7, it is evident that all number of all the intermediate parts and supply parts remain steady over time, but we again encounter the problem of surplus stock of intermediate parts P_7 and P_4. This can be attributed to the higher rate of production of parts P_4 and P_7, $\lambda_1 = 8$ and $\lambda_4 = 8$ respectively, during the initial period of simulation. Based on the design of the supply chain, P_3 is consumed to produce P_5, and P_5 is further consumed to produce P_7. We can, thus, adjust the initial supply of part P_3 to mitigate challenge of the surplus amount of P_7. Similarly, initial supply of P_2 can be tuned to deal with the initial surplus production of P_4. We assume the same parameters for safety-stock strategy and simulate the supply chain with the updated initial state: $\{500, 390, 360, 0, 0, 0, 0, 0\}$ in Figure 3.8.

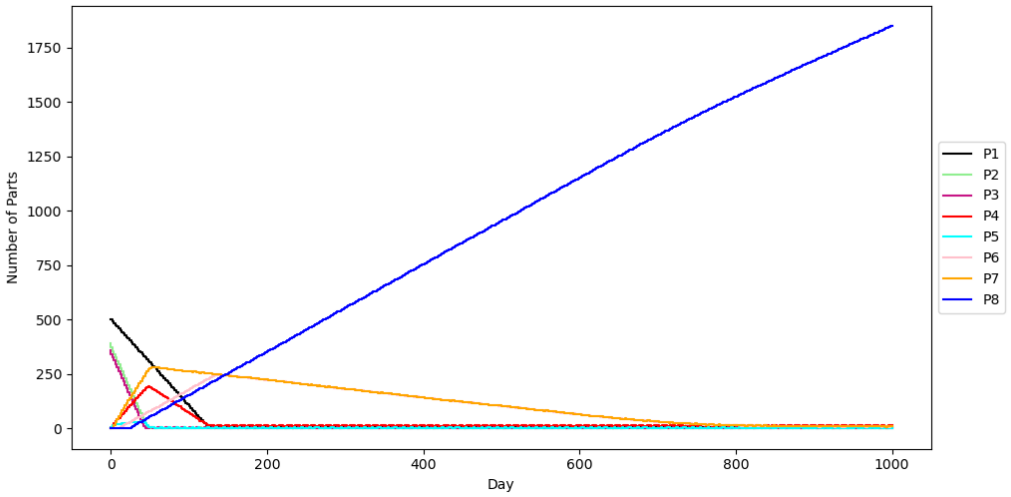


Figure 3.8: Time histories of the supply chain represented in Figure 1.1 after incorporating safety-stock strategy, aggregated over 100 simulations, $\tau = 2$ days, simulated over 1000 days. Initial state: $\{500, 390, 360, 0, 0, 0, 0, 0\}$. Safety stock for supply parts P_1, P_2 and P_3 is set to 16.

In Figure 3.8, we appear to have achieved the optimal initial state and values of safety stocks, safeguard orders for a given back-order time, that ensure uninterrupted production of output-part at a steady rate and avoid overstocking and surplus stocks throughout the simulation period of the example supply chain.

3.6 Disruptions in Supply Chains

Disruptions are events in a process that can completely bring down the production of a process for some amount of time. Disruptions are modelled as local events in processes, using two-state continuous time Markov chains with states 0 and 1, where 0 represents the state when the process is going through a disruption event, whereas 1 is the normal state of the process. If a process $j \in \{j \mid 1 \leq j \leq J\}$ goes into state 0 at time t, the consumption and production of parts by process j stops completely until the process again goes into the normal state. Hence, if a process j is in a disrupted state at time t, then $\lambda_j(t) = 0$.

The rate of disruption events for a process j is controlled by a matrix Q_j given as follows:

$$Q_j = \begin{pmatrix} q_{j;0,0} & q_{j;0,1} \\ q_{j;1,0} & q_{j;1,1} \end{pmatrix}$$

where $q_{j;0,0} = -q_{j;0,1}$ and $q_{j;1,1} = -q_{j;1,0}$. The transition between states 0 and 1 are is modelled by simulating the times to transition to next state, also known as sojourn times, using the elements of matrix Q_j. This is done as follows: if the process j transitions to a state a at

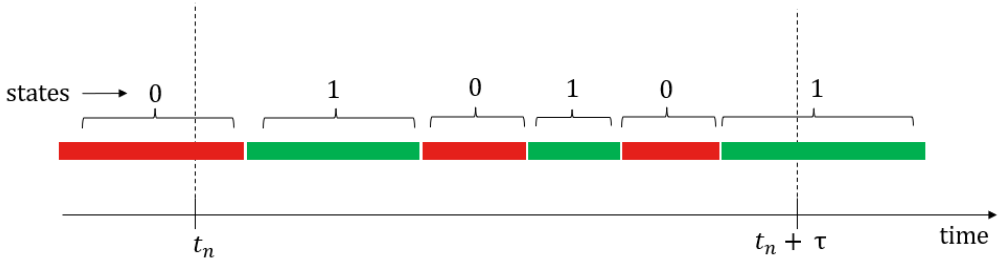


Figure 3.9: Incorporating disruption events simulation into time-bucket algorithm. The red states represent disrupted state and green state represents normal state of the process.

time t, then the time to transition to next state is sampled from an exponential random variable E_a where E_a $exp(1/q_{j;a,b})$ such that $a \neq b$. Thus, higher the value of $q_{j;1,0}$, lesser the time the process takes to transition to the disrupted state, and hence, higher the proportion of time it spends in the disrupted state.

Disruption events in a process j are simulated as follows: the process j starts in an undisrupted state, i.e. state 1 at $t = 0$. We sample w_1, the transition time to the disrupted state 0, from an exponential random variable with rate $q_{j;1,0}$. The process stays in the undisrupted state for w_1 amount of time and at $t = w_1$, the state of process is updated to 0. Now, we sample w_2, the transition time to the normal state 1 from exponential random variable with rate $q_{j;0,1}$, and the state is updated to 1 after spending w_2 amount of time in the disrupted state. These steps are repeated until the time limit of the simulation is reached. All the transition times and state history of the process is recorded, so that it can be incorporated in the simulation of supply chains in the time-bucket method. For a time bucket $(t_n, t_n + \tau]$, let $\delta D_{j,t_n}, 0 \leq \delta D_{j,t_n} \leq 1$ be the proportion of time the process j spends in the undisrupted state, i.e. state 1. The rate of the process j in time bucket $(t_n, t_n + \tau]$ is then calculated as follows:

$$\lambda_{j,t_n} = \lambda_j^{max} \delta D_{j,t_n},$$

where λ_j^{max} is the maximum production rate of process j. The number of occurrences of process j in time bucket $(t_n, t_n + \tau]$ is, thus, sampled as the Poisson random variable with rate equal to $\lambda_{j,t_n}\tau$.

The steady state of the CTMC gives the limiting distribution π of the system, which indicates the portions of time spent by the process in each state:

$$\pi = \left[\frac{q_{j;1,0}}{q_{j;1,0}+q_{j;0,1}} \quad \frac{q_{j;0,1}}{q_{j;1,0}+q_{j;0,1}} \right]$$

The limiting distribution of disruption events in processes can be estimated from observed data, such as shipping logs, port records etc., which in turn can be used to estimate $q_{j;1,0}$ and $q_{j;0,1}$.

3.6.1 A basic supply chain with disruptions

We present a simple supply chain with one process, that represents import of a supply from one destination to other. The delay of the import process is assumed to be deterministic and equal to $t_1^{min} = t_1^{max} = 10$, while the maximum rate of the process is assumed to be $\lambda_1 = 10$. The initial supply of the system in the simulation is taken to be $x_1(t = 0) = 1000$. Let us consider the following Q matrix to control rate of disruptions in the process:

$$Q_j = \begin{pmatrix} -0.5 & 0.5 \\ 0.1 & -0.1 \end{pmatrix}$$

We perform 500 MC simulations for three scenarios:

1. Simulating supply chain without disruption.
2. Simulating supply chain with disruption rate $q_{1,0} = 0.1$.
3. Simulating supply chain with double disruption rate, i.e. $q_{1,0} = 0.2$.

When $q_{1,0} = 0.1$ and $q_{0,1} = 0.5$, the steady state vector of the system is given by $\pi = [1/6, 5/6]$, which indicates that the process spends 16% of the time in disrupted state and 84% of the time in the normal state. On the other hand, when $q_{1,0} = 0.2$, the steady state vector is given by $\pi = [0.28, 0.72]$. Figure 3.10 shows the aggregation of number of imports of 500 MC simulations for the three scenarios described above. It is evident from the plot that there is a significant decrease in rate of imports when disruptions are incorporated in the supply chains. Moreover, increasing the rate of disruptions further decreases the rate of the imports.

3.7 Simulating a global supply chain

In this section, we apply the methodologies mentioned in the previous section to simulate a real supply chain that involves importing parts from different countries, such as China, India and United States of America, to the UK as shown in Figure 3.11.

The supply chain represented in Figure 3.11 represents supply of parts from three countries: China, India and USA. The processes in the supply chain can be divided into two categories: transportation and transit-checkpoints. Transit checkpoints indicate the canals and straits through which the parts pass while en route to being imported into the UK. The range of time-delays of transportation processes should be estimated based on the time taken to transport goods between the respective locations. For example, time delays in process $T_{S\to D}$, i.e. transportation from Suez canal to Strait of Dover are expected to be greater compared to those in $T_{D\to UK}$., i.e. transportation from Strait of Dover to UK. This is due to the fact that goods need to cover a longer distance during transportation between the Suez Canal and the Strait of Dover compared to the distance between the Strait of Dover and the UK. On the other hand, time-delays of straits and canals should be estimated based on the time spent by ships in these

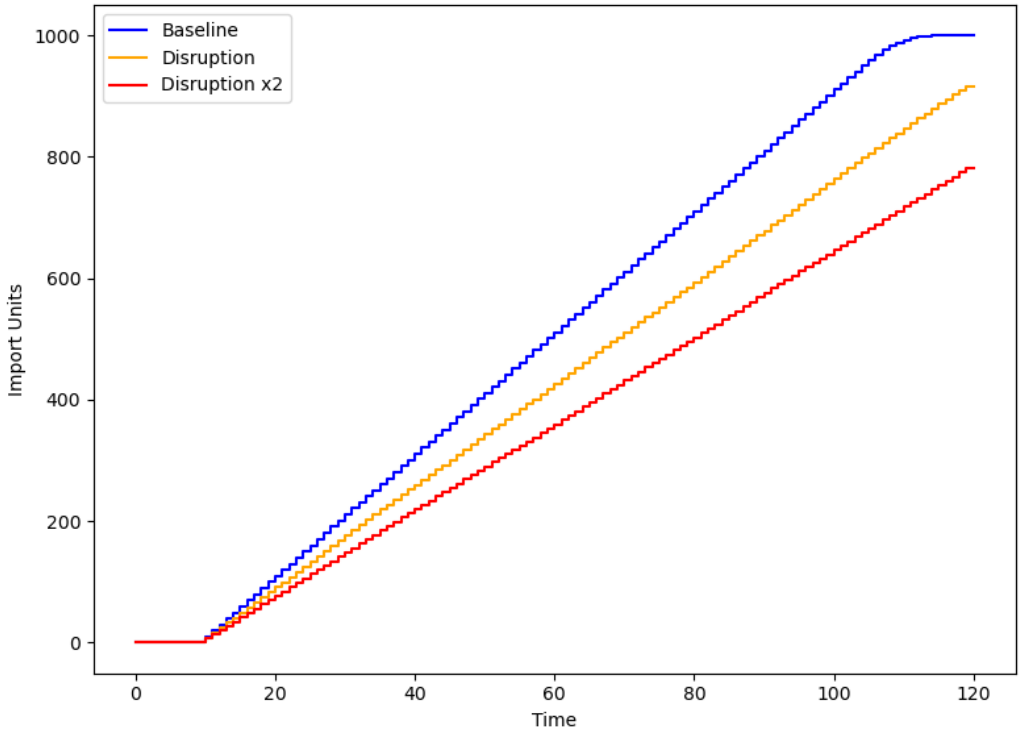


Figure 3.10: Comparison of supply chain dynamics for 500 simulations without disruption (blue), and with two frequencies of disruption events: a baseline level of disruption (orange) and with disruption events that are twice a frequent (red). For the simulations without disruption (blue) the supply (of 1000 units) is fully consumed by 110 time units, as implied by the flattening of the curve.

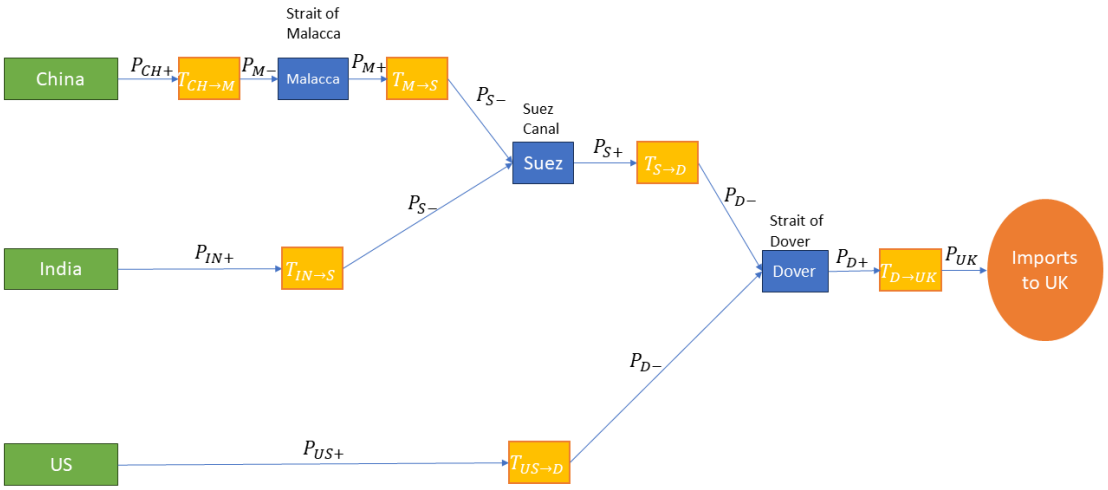


Figure 3.11: A global supply chain with supplies exported from India, China and USA, and imported in UK. The parts are labelled as P_{L+} or P_{L-} where the subscript 'L' indicates the departure or arrival location and the sign '$+$' indicates departure whereas '$-$' indicates arrival of goods. Processes in yellow represent transportation of parts between two locations and are denoted 'T' along with a subscript indicating the departure and arrival locations. Blue processes indicate straits and canals through which the goods are transported, and are named accordingly.

locations while transiting through these checkpoints. The process labels, their descriptions and their corresponding time-delay ranges are mentioned in Table 3.1.

The supply chain represented in Figure 3.11 contains 10 parts, where P_{CH+}, P_{IN+} and P_{US+} are supplies exported from China, India and USA respectively, $P_{M-}, \ldots, P_{D+}$ are intermediary parts and P_{UK} are parts that are imported in the UK. Parts are labelled as P_{L+} and P_{L-} where 'L' indicates the location of the part in the supply chain, and the signs $+$ and $-$ represent where the part has been departed from the location or has arrived at the location L. For example, P_{S-} represents the parts that have arrived at Suez canal and are waiting to transit through the canal, whereas P_{S+} represents the parts that have already passed through Suez canal an are waiting to be transported to different location.

We simulate the supply chain with following transportation rates.

$$\lambda_{T_{CH\to M}} = \lambda_{T_{IN\to S}} = \lambda_{T_{US\to D}} = \lambda_{T_{M\to S}} = \lambda_{T_{S\to D}} = 100,$$

$$\lambda_{T_{D\to UK}} = 200.$$

The process rates of transit-checkpoints are given as follows:

$$\lambda_{Malacca} = 100,\ \lambda_{Suez} = 50,\ \lambda_{Dover} = 200.$$

Process name	Process Description	Time-delay range (in days)
$T_{CH \to M}$	Transportation from China to Strait of Malacca	[5, 7]
$T_{IN \to S}$	Transportation from India to Suez Canal	[7, 9]
$T_{US \to D}$	Transportation from USA to Strait of Dover	[7, 9]
$T_{M \to S}$	Transportation of China parts from Strait of Malacca to Suez Canal	[9, 11]
$T_{S \to D}$	Transportation of parts from Suez Canal to Strait of Dover	[5, 7]
$T_{D \to UK}$	Transportation of all parts from Strait of Dover to UK	[0, 1]
$Malacca$	Transit of all parts in Strait of Malacca	[0, 1]
$Suez$	Transit of all parts in Suez Canal	[0, 1]
$Dover$	Transit of all parts in Strait of Dover	[0, 1]

Table 3.1: Supply Chain Processes and their corresponding time-delay ranges

The rates of transportation between different locations is set to 100 based on the assumption that each of the 3 countries (China, India and US) exports on average 100 parts from their ports to the UK. On the other hand, process rates of transit checkpoints such as Strait of Malacca and Strait of Dover are based on the number of ships they allow to pass in a given time period. It is worth noting that Suez canal has a relatively lower capacity than strait of Malacca and Strait of Dover, as it can allow only 50 ships to pass in one day. Assuming an initial state of $x_{CH+}(t=0) = 10000, x_{IN+}(t=0) = 10000, x_{US+}(t=0) = 10000$, the time histories of all the parts as shown in Figure 3.12.

The dynamics of the supply chain are effectively captured in Figure 3.12. Firstly, it is evident that the rates of increase of parts P_{S-} and P_{UK} are approximately equal and close to $150\ parts/day$. The increase and accumulation of parts P_{S-} is a consequence of the lower capacity of Suez canal ($\lambda_S = 50\ parts/day$) when compared to transportation rates from Strait of Malacca and India ($100\ parts/day$). Parts arrive at the Suez canal at an average rate of $200\ parts/day$ and are departed from the canal at a rate of $50\ parts/day$, hence resulting in accumulation of parts P_{S-} at the canal. On the other hand, parts arrive at the Strait of Dover at a rate of $50\ parts/day$ from the Suez canal and $100\ parts/day$ from the US, hence explaining the approximate import rate of $150\ parts/day$ in the UK.

To prevent the undesired accumulation of parts P_{S-} at the Suez Canal, adjustments must be made to the initial state of the supply chain. This can be achieved by reducing the initial number of shipments from India and China. Moreover, we notice a decrease in the import rate to the UK to $0\ parts/day$ after a duration of $400\ days$. This emphasizes the necessity of implementing a safety stock strategy to guarantee a steady flow of parts to the UK.

Transportation of parts from China and India to the UK requires the parts to pass through the Suez canal. It is observed from the process rates of transportation and transit-checkpoints in the supply chain that the capacity of Suez canal acts as a limiting factor in the import rate of parts

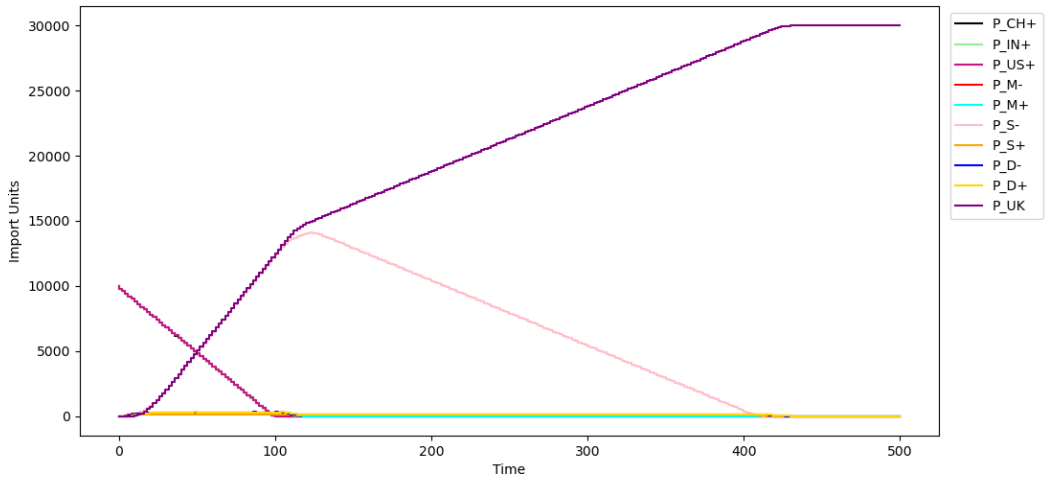


Figure 3.12: Time histories of all parts in supply chain shown in Figure 3.11 aggregated over 50 simulations simulated over a period of 500 days, with $\tau = 2\ days$.

from China and India to the UK. Hence, at each stage in the supply chain, the parts exported from China and India should be transported and transit at an average rate of 50 $parts/day$ in order to avoid accumulation of parts at the Suez canal. We can divide the rate equally amongst India and China, so that the combined rate of arrival of parts from India and China to Suez canal adds up to 50 $parts/day$ on average.

Let us assume that all three countries take $t^b_{CH+} = t^b_{IN+} = t^b_{US+} = 10$ days to prepare a new batch of parts to be exported to the UK via the supply chain. Let us allocate 25 $parts/day$ of export rate to China and 25 $parts/day$ of exports to India, so that the combined arrival rate of parts in the Suez canal is 50 $parts/day$ on average. Taking into account the back-order time of 10 days and the time-bucket $\tau = 2\ days$, let us set the safety-stock and safeguard-orders such that China prepares 25 $parts/day$ on average for export. Thus, we set the safety stock and safeguard order for China as follows:

$$\begin{aligned} x^s_{CH+} = S_{CH+} &= 6\ (5\ timebuckets\ for\ order\ arrival + 1\ timebucket\ for\ placing\ the\ order) \\ &\quad \times 50(average\ parts\ to\ be\ exported\ per\ timebucket) \\ &= 300. \end{aligned}$$

Similarly, we set $x^s_{IN+} = S_{IN+} = 300$.

In case of the US, the exported parts do not have to transit through Suez canal, and hence parts can be exported in accordance with its maximum capacity. Hence, we set the safety stock

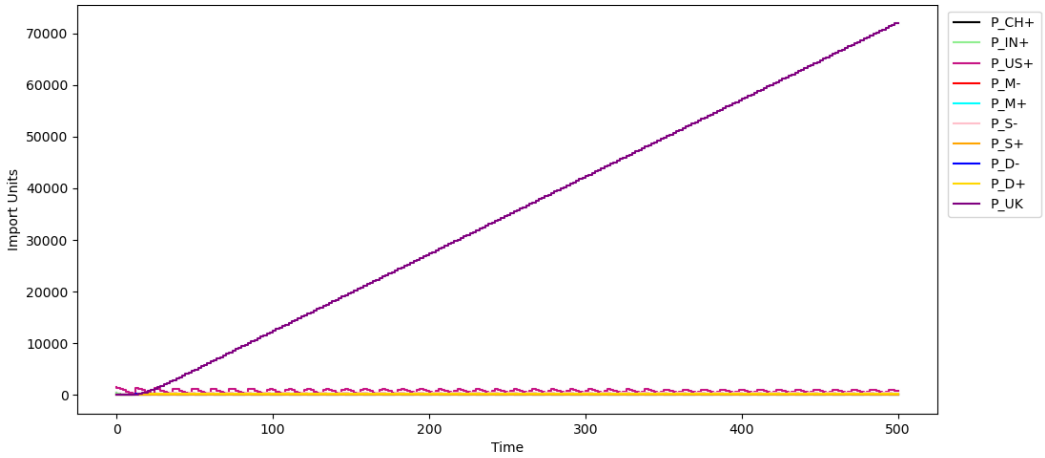


Figure 3.13: Time histories of all parts in supply chain shown in Figure 3.11 aggregated over 50 simulations simulated over a period of 500 days, with $\tau = 2\ days$.

and safeguard-orders for the US as follows:

$$\begin{aligned} x^s_{US+} = S_{US+} &= 6\ (5\ timebuckets\ for\ order\ arrival + 1\ timebucket\ for\ placing\ the\ order) \\ &\quad \times 200(average\ parts\ to\ be\ exported\ per\ timebucket) \\ &= 1200. \end{aligned}$$

With the above values of safety stocks $\{x^s_p\}$, safeguard-orders $\{S_p\}$ and back-order times $\{t^b_p\}$, and adjusting the initial state to $x_{CH+} = x_{IN+} = 1000, x_{US+} = 10000$, we incorporate the safety stock algorithm (Algorithm 5) into the time-bucket method and simulate the supply chain. The time-histories of the supply chain parts are shown in Figure 3.13.

Disruptions in UK supply chain

One of the most important aspects of simulating global supply chains is observing the impact of disruptions on transport rates and import rates. The focal points of potential disruptions predominantly reside at the transit checkpoints. Canals and straits can encounter various challenges that lead to disruptions in the supply chain. For instance, in the Suez Canal, unexpected blockades can swiftly disrupt the flow of goods, causing delays in the delivery of essential components. Similarly, the Strait of Malacca can be vulnerable to disruptions caused by adverse weather conditions, such as monsoons, which can impede shipping schedules and create unpredicted bottlenecks.

Moreover, transportation routes over seas can also face disruptions due to extreme weather conditions or political tensions. Although, these disruptions are less likely than disruptions in canals and straits.

We incorporate disruptions in the supply chain shown in Figure 3.11 using the methodology

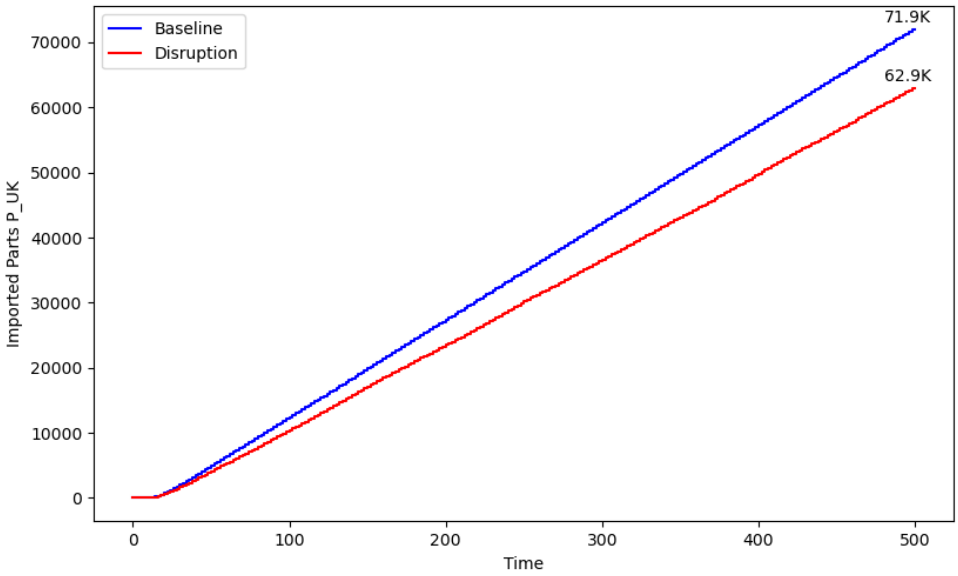


Figure 3.14: Plots showing number of imported parts in the UK within 500 days before and after incorporating disruptions, with with disruption rates given by (3.4) and (3.5).

mentioned in Section 3.6. We use the following rate matrices $\{Q_p\}$ to describe the rates of disruption of each process:

$$Q_{\text{Transportation}} = \begin{pmatrix} -0.5 & 0.5 \\ 0.01 & -0.01 \end{pmatrix} \tag{3.4}$$

Equation (**??**) shows the matrices for Dover and Suez:

$$Q_{\text{Dover}} = Q_{\text{Suez}} = Q_{\text{Dover}} = \begin{pmatrix} -0.5 & 0.5 \\ 0.2 & -0.2 \end{pmatrix} \tag{3.5}$$

We recall that for a process p, a higher value of $q_{p;1,0}$ corresponds to a shorter transition time to the disrupted state. Consequently, the process spends a higher proportion of time in the disrupted state.

In order the compare the impact of disruptions on the import rate of parts P_{UK} to the UK, we make two plots:

1. Number of parts P_{UK} imported in the UK without incorporating disruptions.

2. Number of parts P_{UK} imported in the UK after incorporating disruptions using rate matrices mentioned in (3.4) and (3.5).

The two plots are shown in Figure **??**. It is evident from the plots that disruptions impede the rate of import of part P_UK. On average, only 62900 units of P_UK are imported within 500

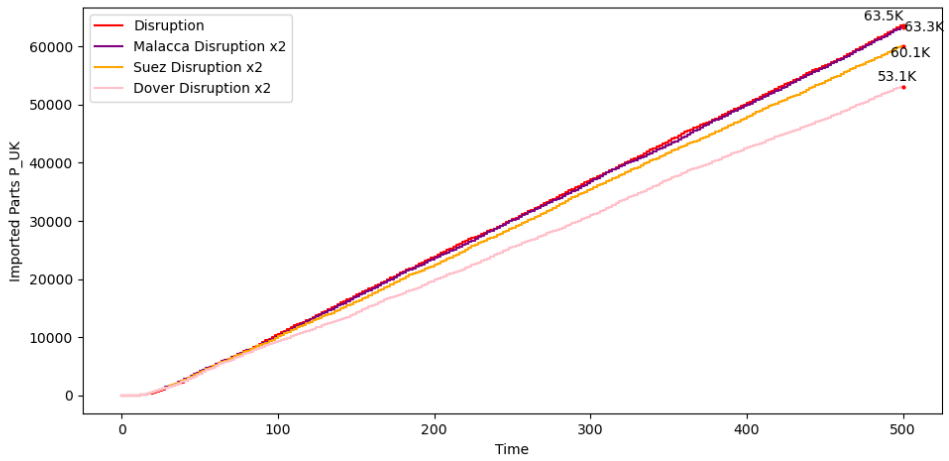


Figure 3.15: Plots showing number of imported parts in the UK after incorporating disruptions with rates mentioned in (3.4) and (3.5), and doubling the value of $q_{p;1,0}$ for Suez, Malacca and Dover.

days to the UK if the processes in the supply chain are made to undergo disruptions, whereas 71900 units are imported if all the processes operate seamlessly.

It is often helpful to assess the impact of disruptions in each of the processes on the import rates. This assessment can be helpful in designing efficient supply chains and drafting appropriate policies. In the supply chain shown in Figure 3.11, the transit-checkpoints have higher chances of undergoing disruptions, and hence, it becomes crucial to examine the impact of disruptions on import rates. We simulate the supply chain with four plots, as mentioned below:

1. Number of parts P_{UK} imported after incorporating disruptions using rate matrices mentioned in (3.4) and (3.5).

2. Number of parts P_{UK} imported after incorporating disruptions, with $q_{Malacca;1,0} = 2 \times 0.2 = 0.4$, and rest of the values same as (3.4) and (3.5).

3. Number of parts P_{UK} imported after incorporating disruptions, with $q_{Suez;1,0} = 2 \times 0.2 = 0.4$, and rest of the values same as (3.4) and (3.5).

4. Number of parts P_{UK} imported after incorporating disruptions, with $q_{Dover;1,0} = 2 \times 0.2 = 0.4$, and rest of the values same as (3.4) and (3.5).

From Figure 3.15, we see that disruptions in Strait of Malacca have the least impact on import rates. Based on the design of supply chain in Figure 3.11,the disruptions in Strait of Malacca are expected to impact only the exports coming from China. Moreover, as only 25 $parts/day$ are allotted to China, very little change in import rate is observed when compared to the impact from original values of disruptions in (3.4) and (3.5). On the other hand, disruptions in the Suez Canal have a greater effect on import rates due to the requirement for parts from both India and

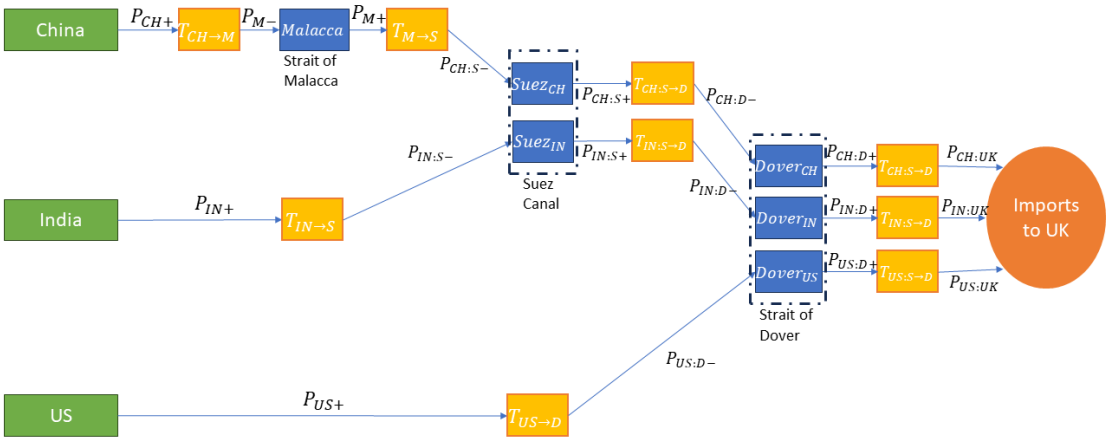


Figure 3.16: A global supply chain with supplies exported from India, China and USA, and imported in UK. The parts are labelled as P_{L+} or P_{L-} where the subscript 'L' indicates the departure or arrival location and the sign '$+$' indicates departure whereas '$-$' indicates arrival of goods. Sub-scripts of the process names and part names are prefixed with 'CH', 'IN' and 'US' to remove ambiguity, wherever necessary.

China to pass through the Suez Canal before being imported into the UK. Disruptions in the Strait of Dover carry an even more significant impact on the import rate, given that parts from all three countries need to traverse the Strait of Dover before they are finally imported into the UK.

We observed the effects of overall disruptions in the supply chain, as well as pronounced disruptions in specific transit points on the import rate of P_{UK} in the UK. It is often useful to simulate the number of imports arriving the UK from each country separately, that traverse through a common supply chain. Keeping track of parts from each country throughout every phase of the supply chain can facilitate the identification of crucial factors that contribute to inefficiencies, as well as pinpoint sources of cost escalation and delays in the timeline. To achieve this, the supply chain in Figure 3.11 is modified by adding country-specific transit and transportation processes in the supply chain. The modified supply chain is shown in Figure 3.16.

There are 15 processes in the supply chain showed in Figure 3.16, as compared to 9 in Figure 3.11. Each process handles parts exported from only one country, and hence, the process labels are sub-scripted with the abbreviations of the corresponding country names. For example, $Suez_{CH}$ represents the process of transit of parts exported from China in the Suez Canal, whereas $Suez_{IN}$ represents transit of parts from India.

In the Figure 3.17, we try to observe the impact of pronounced disruptions in Suez canal on the number of parts imported in the UK from each country separately. We use the same values

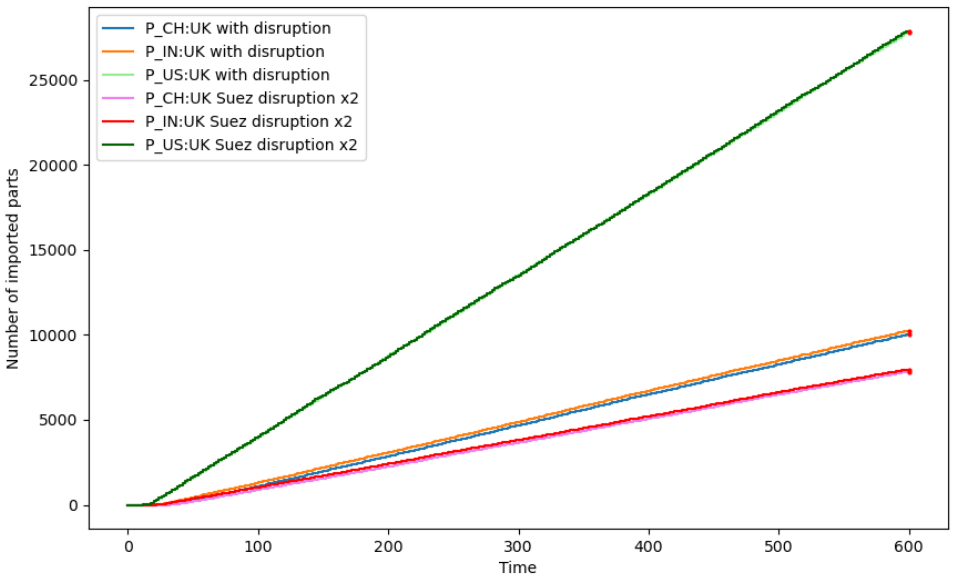


Figure 3.17: Time-histories of parts imported in the UK from India, China and USA, with disruption rates as mentioned in (3.4) and (3.5) and with enhanced disruption in Suez Canal with $q_{Suez;1,0} = 2 \times 0.2 = 0.4$.

of process rates, disruption matrices, safeguard-orders, safety-stocks, back-order time delays as used previously, with $q_{Suez;1,0} = 2 \times 0.2 = 0.4$.

Based on the design of the supply chain in Figure 3.11, we can infer that disruptions in Suez canal should only affect the rates of import of parts from India and China. It is evident from the plots in Figure 3.17 that import rate of parts from the US are unaffected by the disruptions in Suez canal, whereas import rates of parts from both India and China have been affected.

Bibliography

Antoniadis, A., Gregoire, G., & McKeague, I.M. (1994). Wavelet methods for curve estimation. *Journal of the American Statistical Association* **89**, 1340-1353.

Armitage, P. (1998). Attitudes in clinical trials. *Statistics in Medicine* **17**, 2675-2683.

Chatfield, C. (1996). *The Analysis of Time Series, 5th edition*. London: Chapman & Hall.

Malyshev, V.A. (2002). Table of contents to Markov Processes and Related Fields. `http://www.math.msu.su/~malyshev/abs02.htm`, accessed 13 October 2008.