CONCURRENT MODELING OF CAUSALITY, TEMPORALITY, AND

SYSTEMIC RISK IN SUPPLY CHAINS: A HYBRID APPROACH

By

KE SUN

A dissertation submitted to the

Graduate School-New Brunswick

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Industrial and Systems Engineering

Written under the direction of

Dr. James T. Luxhøj

And approved by

____

____

____

____

New Brunswick, New Jersey

October, 2017

ABSTRACT OF THE DISSERTATION

CONCURRENT MODELING OF CAUSALITY, TEMPORALITY, AND SYSTEMIC RISK IN SUPPLY CHAINS: A HYBRID APPROACH

by KE SUN

Dissertation Director:

Dr. James T. Luxhøj

The uncertainty of operations for supply chain involved companies is becoming more complex with the growth of globalized business collaboration. A supply chain is a complex system with dynamic flows of capital, goods, information and people. The temporal fluctuations of politics, economics, nature and technology on a supply chain process may potentially cause disruptions to the whole system. The malfunctions of supply chain systems around the world cost companies billions of dollars and months of recovery time every year. Prevention and mitigation of supply chain disruption risks are crucial for companies to maintain their competitive advantage. However, the performance of mitigation plans may be unsatisfactory due to a myriad of interactive impact factors in supply chains under uncertainty. This situation requires a direct and concise tool to monitor and control supply chain risks concurrently.

Various qualitative and quantitative risk analysis tools are introduced to unveil the myth of uncertainty. A Bayesian Belief Network (BBN) is one of the risk modeling approaches that provides a systemic conditional probabilistic view on risk analysis. A Dynamic Bayesian Network (DBN) offers a solution that enables temporal factors in a BBN, which is in consonance with the characteristics of time-sensitive risks in supply chains. Inputs and outputs of DBNs are probability values. Supply chain practitioners may have difficulty to concretize the values into practical operations immediately because supply chain performance is measured with actual units of money or inventory. System Dynamics (SD) is a simulation tool for modeling complex socio-technical systems in feedbacks, stocks and their flows. However, SD has limitations in simulating conditional probabilities within the dynamic flows.

By utilizing the essence of DBN and SD, this dissertation proposes a Dynamic Flow Bayesian Network (DFBN) to offer a comprehensive methodology for supply chain risk analysis. An Optimized Dynamic Flow Bayesian Network (ODFBN) method is developed with modifications based on the DFBNs by incorporating multi-objective optimization, multi-pricing strategy and Value-at-Risk. By applying the concept of Supply Chain Network Equilibrium, an Equilibrated Dynamic Flow Bayesian Network (EDFBN) method is developed to balance the needs of each stage and maximize the profitability of the entire supply chain. In this dissertation, mathematical integration of the models is presented and application to a supply chain case study inspired by the real-world is also conducted. Finally, a prototypical executable interface for industrial implementation is developed.

ACKNOWLEDGMENTS

I would like to present my greatest gratitude to my Ph.D. advisor Dr. James T. Luxhøj, who gave me not only his full support on academic research but also his guidance on the way of thinking, the philosophy of decision-making and the importance of family. Dr. Luxhøj provided me an irreplaceable research experience that enlarges my vision and enhances my attitude on dedication. The dissertation committee members Dr. Mohsen A. Jafari, Dr. Sheila M. Lawrence and Dr. Myong K. Jeong make a significant contribution to the completion of this research. I would like to thank them for their generous efforts and sincere suggestions.

The financial support of this research is provided by Dr. Mohsen A. Jafari, the Department of Industrial and Systems Engineering of Rutgers University and my parents. This research would not possibly be done without their kindly help.

My special thankfulness goes to my parents Chengyi and Jingping. Their warm encouragement is always there when my confidence of advancing is challenged.

Last but not least, my significant other, Chuyin, your unwavering love brightened my journey to one of the major achievements in my life. I believe that with the faith we have, nothing will cease our life-long romantic partnership.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

1. Introduction: Risks and Supply Chains

Change is a constant through human history. Our world is transformed by accelerating changes in population, economic activity and technology (Sterman, 2000). Some changes, for example, such as the application of information technology, improve people's lives, but others such as global warming, may deteriorate our living environment. Human activities create most of the changes, but not all of the results are consistent with people's original expectations. The gap between the expectations and actual outcomes is uncertainty. Occasionally, surprising positive outcomes may be generated by an uncertain operating circumstance. However, business practitioners with a sense of crisis prefer a hypothetical negative outcome of their decisions when operating in uncertain circumstances.

Various risk analysis tools are developed to understand and mitigate risks and uncertainty. With the probing of risk behaviors, the complex relationship of risk factors in systems is gradually realized. Thus, more organized and systemic models are developed to present risks in complex systems. A perfect emulated risk model in complex systems may be unrealistic due to the limitation of existing tools. For software, a mathematical expression that describes risks in complex environments with complete accuracy can hardly be achieved (Keynes, 1936). For hardware, the computing technology that can simulate an accomplished virtual reality risk environment is inaccessible. However, a number of applications of systemic risk analysis methods are developed with continual efforts.

A supply chain is a typical complex system. All parties that directly or indirectly involve satisfying a customer request constitute a supply chain (Chopra & Meindl, 2010). The parties involved may include manufacturers, suppliers, retailers, warehouses, transporters and customers. Supply chains consist of the dynamic flows of capital, product, information and people. All functions in an organization that relate to receiving and filling a customer request are included in the supply chain, such as product research and development, marketing, finance, operations, distribution and customer relationship management.

Organizations and their functions are connected through supply chains. One action taken in one division may trigger a chain effect to a significant number of its upstream and downstream parties, especially in a globalized business environment. Thus, the smooth operation of supply chains is critical to the liquidity of flows between organizations. Failures in the operations are called disruptions in supply chain systems. Uncertainties from the external environment (e.g., political, natural or social uncertainties) and the internal environment (e.g., supply, demand or manufacturing uncertainties) potentially evoke supply chain disruptions (Jüttner, 2005). The disruption may yield a hazardous impact to related industries every year. In 2011, severe floods defeated industries in Thailand, which is a global supply chain hub located in Southeast Asia. Thousands of businesses were affected, including international manufacturing companies such as Apple, Toyota, Honda, Ford, Sony and Toshiba. The impact dismantled their global supply chain strategies, especially for the world's largest hard-disk drives supplier Western Digital. The floods suspended about a quarter of the hard-disk drive supply, and Western Digital took months to resume normal production. The top five supply chain disruptions in 2014 caused more than 17.3 billion U.S. dollars' loss. In average, it takes 31 weeks for the industries to recover from each impact. Three out of five of these disruptions occurred in Southeast Asia. The loss is equivalent to approximately 3.3% of the gross domestic product (GDP) of Thailand in 2014.

A supply chain involves a substantial amount of livelihood issues for individuals and organizations. The dynamic flows in a supply chain fluctuate arbitrarily. The pattern of operation may be abruptly altered by an incident and cost millions of dollars to restore. A reliable supply chain risk analysis model, which considers temporality and resource flows, can contribute to a variety of industries to minimize the impact of their business uncertainties.

By combining the essence of Dynamic Bayesian Networks (DBNs) and System Dynamics (SD), this research proposes a Dynamic Flow Bayesian Network (DFBN) to offer a comprehensive methodology for supply chain risk analysis. For probabilistic risks, DBNs provide a systemic analysis tool to dissect risk factors from an intricate circumstance. Feedback loops in SD models utilizes probabilistic risk values from DBNs to generate information flows. Combined with other physical and financial flows in SD, systemic risks can be concretized into actual capital or inventory outputs for supply chain decision-making. The modifications to the DFBN is also provided in this dissertation.

This chapter provides an overview on the motivation of the study, along with the limitations of existing models dealing with complex uncertainty. The potential issues or challenges of the existing models as applied to the supply chain problem domain are presented. The research problem is defined and objectives are proposed to offer a temporal probabilistic causal modeling solution for the identified problems.

1.1. Systemic Complex Risks

The globalization of business activities not only offers opportunities for companies to achieve greater benefits from international collaboration, but also yields potential threats from decisions in other parts of the world. The business environment grows into a more complex pattern. It requires business practitioners to expand their cognition of risks into systemic thoughts.

1.1.1. Risks

- What Are Risks?

The business world is replete with uncertainty. However, uncertainty is a notion without a straightforward description (Antunes & Gonzalez, 2015). As the first economist to bring uncertainty as a distrust factor to economics, Keynes (1936) established the specification of uncertainty using statistics to provide guidance for rational actions in an uncertain world.

Uncertainty is a potential and uncontrollable outcome, and risk is the consequence of humans' actions taken in such environments (Mun, 2006). In an uncertain environment, risks exist in an endeavor in business and industrial projects. In other words, the actions that a decision maker takes may affect the outcome or the performance of a business. Although uncertainty and risk are ever present, it is possible to mitigate, transfer, or even avoid the impact of risk through the identification, assessment, and formulation of operation plans (Crouhy et al., 2006). Efforts have been taken by both practitioners and scholars to understand and mitigate the influence of these risks. The actual value of capital affected by risk is factored by the odds of the risk occurring and the value of the impact, and risk mitigation plans may focus on these two factors.

- What Are Systemic Risks?

In the real world, complexity is the natural property of risk and uncertainty. The cause-effect relations are embedded in every action that a creature takes in nature. In 1961, a computer weather prediction was conducted by Edward Lorenz. When redoing a shortcut simulation from the middle of the previous run, he entered 0.506 instead of 0.506127 as one of the initial values. However, the result indicates a completely different weather scenario (Lorenz, 1963). The term "butterfly effect" that describes this phenomenon was then summarized from a presentation of Lorenz (1972): "predictability; does the flap of a butterfly's wings in Brazil set off a tornado in Texas?" The butterfly effect shows an extraordinary but possible outcome of these complex relations hidden behind the connections behind objects. A systemic relation view on uncertainty and risk may provide a comprehensive perspective to business models. Systemic risk is originally a financial expression. It refers to "the propagation of an agent's economic distress to other agents linked to that agent through financial transactions" (Rochet, 2003). The expression elaborates the spreading of risks between two business parties. If we place the propagation in a larger system, such as a supply chain, the risk of one decision in one stage may cause a chain effect of distress to its upstream and downstream industries.

- Why Is Understanding These Risks Important?

Understanding systemic risks in different business scenarios helps decision makers become aware of the potential consequence of their choices during operation. The profit in business is negatively influenced by risk-affected values. A proper understanding of systemic risks ensures that the business goal, which is usually maximizing profit with limited resources, is not deviated by the decisions under uncertainty. A contingency plan is a primary maneuvering tool against risks (Cleden, 2012). It neutralizes the total risk-affected value by both reducing risk occurrence odds and impact. In addition, in projects with a high concentration of events, it is paramount to computing the pattern of dissemination of the contingencies (Antunes & Gonzalez, 2015).

1.2. Probabilistic Risk Assessment Models

Risk assessment models have been established in order to help managerial professionals in different business domains to understand, assess, and probably avoid risks veiled in their industry. Risk assessment models are mainly categorized into two classes: qualitative methods and quantitative methods.

1.2.1. Qualitative Methods versus Quantitative Methods

A qualitative risk assessment method can describe the probability of the occurrence of undesired events by utilizing subjective risk levels (Wieland et al., 2011). The output through a qualitative model is a linguistic expression that presents an approximate image of risk for a reader. The accuracy of the modeling result highly depends on the degree of precision of the language used. There is also a gap between the linguistic expression and the comprehension of humans for that expression, which potentially adds another risk factor to the model. Cox et al. (2005) summarize the limitations of a qualitative method in two aspects: reversed rankings and uninformative ratings. Reversed rankings represent the misclassification of risks in inappropriate levels. Uninformative ratings indicate the errors in risk classification due to the uninformative linguistic level labels.

However, qualitative methods are widely applied in data-scarce environments to provide a useful tool to mitigate the risk for risk managers (Singer, 2010). Linguistic expressions also show strengths in decision communication between practitioners. Qualitative risk assessment methods also possess the ability to investigate systemic risks in a complex environment (Zalk, 2009; Bass & Robichaux, 2001; Wieland et al., 2011).

Despite the concise presentation of systemic risks by using qualitative methods, developing and applying quantitative methods is recommended by Cox et al. (2005) to conduct more reliable risk assessments. According to Kaplan & Garrick (1981), Quantitative Risk Assessment (QRA) basically answers three questions: What can go wrong? How likely is it? And what are the consequences? QRA has been found useful at providing insight on systems in scenarios with multiple failures. QRA also helps business stakeholder groups identify complex interactions between events, systems and operations. Valuable inputs to decisions can be generated by focusing on uncertainty quantification of QRA (Apostolakis, 2004). Various QRAs are introduced to accomplish specific tasks required in different systems or system levels, such as Event Tree Analysis, Fault Tree Analysis, Bayesian Belief Networks, Sensitivity Analysis, Monte Carlo Simulation and Cost Benefit Analysis.

1.2.2. The Application and Limitations of Existing Quantitative Risk Assessment

- Application of QRA

QRA is found to be a valuable tool for risk assessment in various disciplines. Chen & Kodell (1989) present a “benchmark dose” confidence interval assessment method to obtain risk estimates for teratological effects. By combining beta-binomial distribution to litter effects data and a Weibull model to teratogenic data, a lower limit on the safe dose toward reproductive and developmental toxic risks is established to replace the previous statistically insensitive model. Han & Weng (2010) propose a probability risk assessment of natural gas pipelines. Both external risks that are caused by environmental or societal accident and internal risks such as the reliability of the pipeline itself are considered. A promising result for practical application is obtained through a sample urban gas pipeline network assessment. Kolar & Lodge (2002) develop and use multivariate QRA models to assess the risks of alien species invasion to local ecological systems. Alien fish species with high nuisance risk to the Great Lakes are identified with an accuracy range from 87% to 94%.

- Limitations of QRA

With the facilitation of data, QRA acts as an important role in risk studies. However, limitations still exist in QRAs. According to Keynes (1936), a strict mathematical expression on risks in the complex world is unreliable. It makes scholars devote themselves for decades to pursue generations of quantitative models to describe the uncertainty around people. Apostolakis (2004) summarizes several factors that current QRAs are not adequately handled: (1) human omission errors (prescribed actions are not taken), commission errors (improper operations that worsen the situation) and the innovation after errors are not appropriately modeled; (2) dubious assumptions on digital software failures are made; (3) current QRAs can hardly take safety cultures on crew behavior into consideration; and (4) existing QRAs also have limited ability to handle design and manufacturing errors.

1.2.3. The Expectation of Improvements on Existing QRAs

Each QRA tool possesses a certain number of strengths in tackling a range of problems. An omnipotent method may not be realistic with the limitations on both modeling software and hardware. In a complex realistic scenario, complete risk relationships can hardly be expressed by one method alone. Thus, model integration is an approach to include diversified assessment systems and metrics to compensate shortcomings of each single model.

Synergies during modeling and assessing applications can be created by integrating qualitative and quantitative methods. According to McNab & Alvas (2003), in Ontario, Canada, the Ontario

Ministry of Agriculture and Food (OMAF) suggests that quantitative risk assessments are preferred. Nevertheless, quantitative data are often scarce. Thus, OMAF proposes a framework for organizations that aligns qualitative assessments model formalization with quantitative risk. Through this method, risk assessment can be applied in both data sufficient and insufficient scenarios.

The integration between quantitative methods also provides a novel perspective for QRAs in complex system applications. Wang et al. (2008) propose an integrated method that initially uses the analytic hierarchy process (AHP) to determine linguistic expressions on each bridge structure risk criterion and the weight of each criterion. After quantifying the linguistic expressions by using data envelopment analysis (DEA), an aggregate overall risk score for each bridge structure under each criterion is obtained by a simple additive weighting (SAW) method. Through a systematic risk analysis process, the integrated method provides a value for engineers to detect the hazard of each bridge structure from scattered risk data.

The motivation of model integration may be mainly summarized as widening the risk information inputs and outputs availability of each model. In the OMAF case, the qualitative methods enable the application of quantitative methods in an input data-scarce circumstance. In the bridge structure risk case, DEA quantifies the linguistic data generated from AHP, which permits further quantitative data integration.

The efforts taken in risk model integration reflect the pursuit of obtaining more straightforward outputs for decision-making from more precise risk information inputs. For a QRA method, the input variables for risks may usually be represented as probability values. However, if the output of the risk models is a probability value itself, it may be difficult for the analysts to concretize the concept to the real world. In practical applications, especially in dynamic scenarios such as business practices, it is more valuable for decision makers to have straightforward values to assess the impact of risks and their mitigations.

1.3. Supply Chain Risk Analysis

Supply chain risks may originate from every process in the operation. An effective and efficient risk analysis provides a reliable tool for decision makers preventing potential supply chain disruptions or minimizing the loss in catastrophes.

1.3.1. Supply Chain Structures and Processes

According to Angerhofer & Angelides (2000), a supply chain is defined as a production-distribution system with dynamic complex flows of information, materials, orders, money, people and capital equipment. In order to fulfill a customer request, a supply chain consists of all directly or indirectly involved parties (Chopra & Meindl, 2010). Not only suppliers and manufacturers, but also warehouses, transporters, retailers and customers are included in the supply chain. In the globalized business and political environment, achieving the balance between supply chain responsiveness and efficiency supports the competitive strategies of a company. Before understanding the approaches that may best assist the balance, logistical and cross-functional drivers that influence the performance of supply chains should be identified, such as facilities, inventory, information, transportation, sourcing and pricing. When analyzing the decisions and their consequences that relate to the drivers, Chopra & Meindl (2010) suggest that the components and metrics of the drivers should be considered, and they are shown in Figure 1.1.

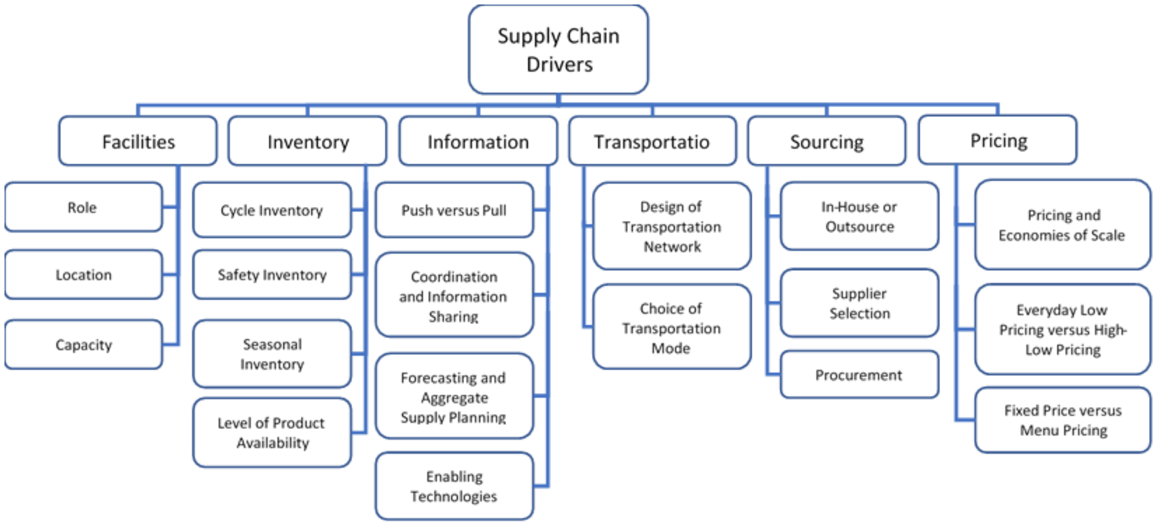


Figure 1.1 The Supply Chain Drivers and Their Components and Metrics.
(Source: Adapted from Chopra & Meindl, 2010)

The interaction of these drivers determines the performance of supply chains in terms of responsiveness and efficiency. In order to realize the profit of the different parties from the separated locations, the dynamic flows of capital, commodity, information and people are formed. During the flows in a supply chain, a turbulent business environment creates risks for the business processes. These risks, which vary over time, may bring negative effects or even catastrophe to the performance of companies in a supply chain. Risk-related operation and mitigation plans are important approaches for practitioners to assess the status of the supply chain. Thus, various supply chain risk analysis methods, both qualitative and quantitative methods, are established to help analysts understand and mitigate those risks and enhance performance.

1.3.2. Existing Supply Chain Risk Analysis Methods

According to Jüttner et al. (2003), supply chain risk is defined as any risk involved in the material, information and product flows during the processes from original supplier to the end user. Supply chain risks can be classified into internal and external risks based on the source of the risks (Wu et al., 2006; Trkman & McCormack, 2009; Olson & Wu, 2010). Supply chain risk analysis focuses on the probability of an event occurring and the consequence significance (Ho, 2015). Moreover, as a complex system, supply chain risk analysis is suggested to be a cross-company process that aims at the identification and mitigation of risks at a macro business environment level (Thun & Hoenig, 2011). Risk analysts mainly apply qualitative methods in supply chain risk identification. Quantitative methods have been developed to analyze and assess the identified risks (Ho et al., 2015).

Both single and integrated quantitative approaches are used in supply chain risk analysis (Ho et al., 2015). Mathematical programming is one of the most singly used quantitative methods, including unconstrained and constrained mathematical programming, linear programming, non-linear programming, integer non-linear programming and stochastic linear programming. Fuzzy logic, AHP and Data Envelopment Analysis (DEA) are the most integrated methods in hybrid models. Compared with the original methods, the integrated techniques can be applied to overcome limitations or strengthen performance. For example, in the integrated method offered by Kumar et al. (2006), fuzzy logic relaxes the requirements of multi-objective mathematical programming on deterministic and exact value identification and generalizes the application of the models on a complex system. By considering the advantages of model synergies, Ho et al. (2015) state that integrated models will attract more attention in the future.

1.4. An Integrated Dynamic Uncertainty Approach for Supply Chain Risk Analysis

In this dissertation, Dynamic Flow Bayesian Network (DFBN) models that integrate a Dynamic Bayesian Network (DBN) with System Dynamics (SD) are proposed to create a systemic supply chain risk model from a fresh perspective. Supply chain risk analysis can be viewed as a model with four factors: systemic probabilistic risks, feedbacks, stocks and their flows. Probability theory plays an important role in describing the likelihood of event occurrence. A Bayesian Belief Network (BBN) is one of the modeling approaches that provide a systemic conditional probabilistic view on risk analysis through a directed acyclic graph. BBN modeling provides a system thinking a probabilistic tool for a complex system with highly correlated risk factors to capture the key systemic risks. A Dynamic Bayesian Network (DBN) offers a solution that enables time dependencies in a BBN (Dagum et al., 1992). The application of DBNs is in consonance with the characteristics of time-sensitive risks in supply chains. However, it is not adequate for a model constrained with the separation of risk models and dynamic flows in a supply chain. System Dynamics (SD) provides a simulation solution for complex socio-technical systems in modeling feedbacks, stocks and their flows. Stocks are accumulated states of the system that enable the system with memory and inertia against changes (e.g., flows). A DBN or SD alone can partially resolve probabilistic risks, feedbacks, stocks and flows modeling problems in a supply chain, respectively. Therefore, a series of integrated models, a Dynamic Flow Bayesian Network (DFBN) and its derived models, are developed to provide a novel opportunity to consider all those supply chain risk factors comprehensively and concurrently.

1.4.1. The Fundamentals and Benefits of Temporality and Dynamic Flow Integration.

The fundamental idea of the integration of DBNs and SD models is the time simultaneity of their variables. The variables in DBNs are updated simultaneously with the variables in the causal loops, flows and stocks in SD models. Certain selected temporal probability data in the DBN can be transferred to the SD model as an input variable. Thus, systemic probabilistic affected feedback and forward flows can be generated.

The DFBN models, including the DFBN, the ODFBN and the EDFBN, provide a complex system risk analysis tool that considers systemic risks from dynamic flows in a supply chain. The DFBN models enable temporal systemic risk factors analysis in a physical supply chain process framework. Concretized results in the supply chain, such as procurement rates, inventory levels, disseminating rates, and investment return rates can be obtained for straightforward decision-making. The outputs of the DFBN models are quantified by the familiar measurement units for supply chain practitioners to have instant reflections on the supply chain status. Such instant reflections may provide effective prevention for the potential supply chain disruptions or the “butterfly effect”.

1.5. Research Objectives

In this research, the main objective is constructing an integrated temporal and dynamic systemic risk modeling method for supply chains. The realization of the objective is divided into four phases: building the theoretical framework for the method, applying the theoretical framework to the supply chain domain, developing improvements and modifications of the original model, and developing a practical prototype for executives.

1.5.1. Develop a Quantitative Analytical Method that Integrates Temporal Systemic Probabilistic Risks and Dynamic Flows

The first phase of this research focuses on constructing a theoretical framework for a Dynamic Flow Bayesian Network (DFBN). Dynamic Bayesian Networks (DBNs) and System Dynamics (SD) are combined to establish the DFBN. A DBN provide a temporal systemic probabilistic risk analysis model. SD models merge the probabilistic risk outputs with other dynamic flows in a complex system to generate a concretized result for decision-making. The outcome of this phase is constructing a bridge between DBNs and SD through mathematical expressions.

1.5.2. Demonstrate the Quantitative Analytical Method with an Application to the Supply Chain Domain

Once the theoretical framework is established, practical supply chain metrics are applied to infuse practical meanings to the method. A supply chain is a complex system with dynamic flows that is vulnerable to risks. By dissecting the systemic risk factors and simulating the risk-affected dynamic flows with a DFBN, the impact of each risk factor on the flows can be detected to prevent potential supply chain disruptions. The outcome of this phase describes the systemic interactions between risks and dynamic flows with a supply chain case study by utilizing a DFBN.

1.5.3. Improve and Modify the Original Risk Analysis Tool for More Application Scenarios

After applying the DFBN in a supply chain case study, several modifications of the original DFBN model are formulated. The improvements aim to strengthen the theoretical validity of the DFBN models to more realistic business environments. The DFBN model is modified systematically to present its flexibility and expandability in practical applications.

1.5.4. Develop Methods to Communicate and Display the Temporal Uncertainty in the Modeling to Executives

Finally, after several improvements and supply chain application of the DFBN models are formalized, the research steps forward to generalize the method to industrial implementation. An executable interface is established that enables users to enter modular supply chain risk information and generate displays for decision-making. The outcome of this phase is establishing an executable prototype interface for executives to analyze supply chain risks.

1.6. Overview of the Thesis

This thesis introduces the motivation and background information about the research in Chapter 1. Relevant quantitative risk analysis methods and the essential knowledge for constituting the model are presented in Chapter 2. The taxonomy of supply chain risk analysis processes is demonstrated in Chapter 3 to clarify the key issues in the supply chain domain. Chapter 4 states the principles in the model establishment and provides preliminary results from a sample DFBN application. The first modification to the original DFBN and its application are also presented in this chapter. In Chapter 5, the network equilibrium concept is integrated with the DFBN to enrich the model with the needs and competitions between the business entities in a supply chain. Finally, the summary, contributions, comparison of recent models, current limitations and future directions of the research are presented in Chapter 6.

2. Literature Survey: Representative Risk Modeling and Uncertainty Methods

In this chapter, a literature survey on representative risk modeling and uncertainty methods is conducted to provide foundations and comparisons to the newly developed risk models in later parts of the dissertation.

2.1. Fault Trees and Event Trees

Fault tree analysis (FTA) and event tree analysis (ETA) both use Boolean logic to determine the risks of event failures contributing to the system failures. However, they have various degrees of differences in graphical illustration, analysis objectives and mathematical presentation.

2.1.1. Fault Tree Analysis

Fault tree analysis is a deductive risk estimation tool that models the occurrence between events (Bedford & Cooke, 2001). FTA is a graphical approach and follows a top-down hierarchical structure. By applying Boolean algebra, FTA illustrates the relationships between different events that could happen in a system (Lindhe et al., 2009). It is built starting from the top event to the bottom until the desired system details are achieved. Basic events, which can be seen as the component failures, are defined as the termination of each logic "branch", and intermediate events are between top events and basic events. These events have the probability to contribute to failure of the whole system, and the top event represents the system failure. Here we illustrate an example to elaborate the mechanism of FTA. Consider a system with three non-repairable components with failure probability p_i $(i = 1,2,3)$ as arranged as Figure 2.1.

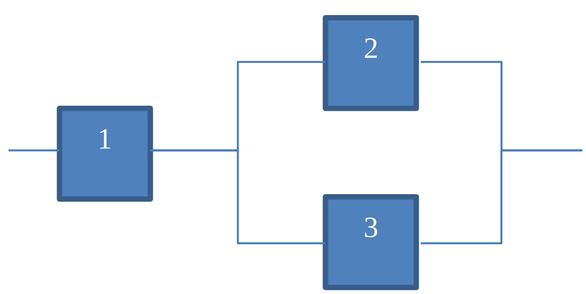


Figure 2.1 A Sample System with Three Components.

A fault tree is constructed to show the failure relationship of the system (Figure 2.2). The failure of the sub-system occurs only if both component 2 and component 3 fail (AND-gate). The probability of a sub-system failure is:

$$P(Sub\text{-}system\ Fails) = P(Component\ 2\ Fails)P(Component\ 3\ Fails) = p_2p_3 \qquad (2.1)$$

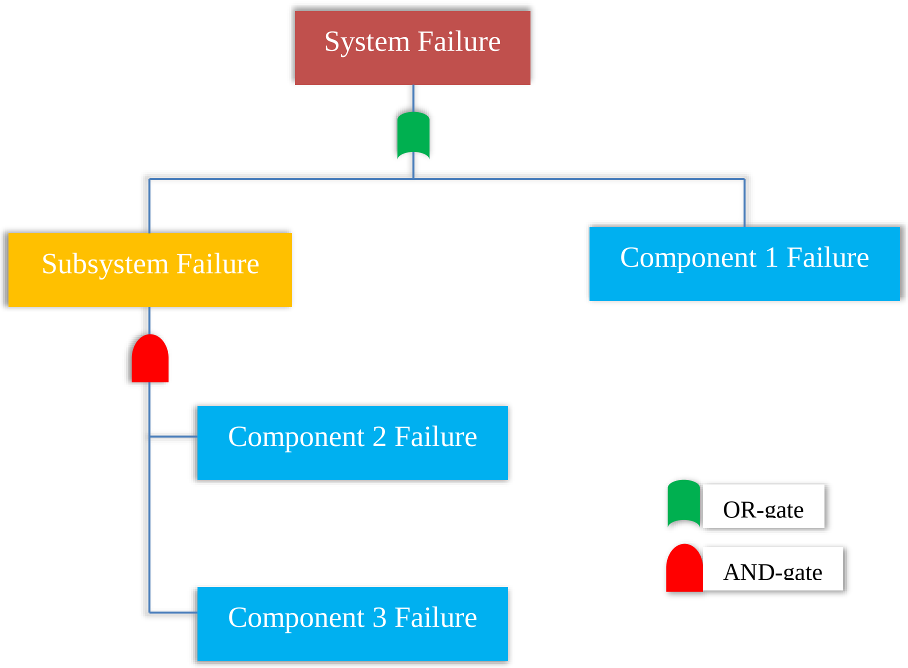


Figure 2.2 FTA Corresponds to the Sample System.

Furthermore, if either the sub-system or component 2 fails, it causes the system failure (OR-gate) with failure rate:

$$P(System\ Fails) = 1 - [1 - P(Subsystem\ Fails)][1 - P(Component\ 1\ Fails)]$$
$$= 1 - (1 - p_2p_3)(1 - p_1) \qquad (2.2)$$

FTA was first introduced in 1962 by Bell Telephone Laboratories for the Minuteman system of the U.S. Air Force (Fussell, 1976). With the straightforward AND/OR logic, an abundant range of system engineering fields, such as nuclear systems (Vesely et al., 1981; Cummings, 1975), railway traffic systems (Hudoklin & Rozman, 1992) and coal mine escape systems (Goodman, 1988) have successful FTA application cases. However, according to Khan & Abbasi (2000), FTA may have weaknesses in representing some systems due to the possible complicated relationships between events; difficulty to collect failure data; time and monetary costs in large scale computation; and unreliable results caused by inaccurate input data.

As classic FTA requires precise failure rates or probability data of components or events, it may not be an appropriate method for industries where such data are inadequate or too large in variation for statistical inferences. In order to extend the application of FTA into such situations, fuzzy set theory, which is developed by Zadeh (1965), is considered to compensate for the lack of uncertainty of FTA. According to the different types of insufficient data obtained in systems, fuzzy set theory is applied to different portions of an FTA, such as the component's failure (Tanaka et al., 1983), the relationship between a component and system failure (Gazdik, 1985), importance measurements (Furuta & Shiraishi, 1984; Tsujimura & Gen, 1994), logic gates (Pan & Yun, 1997), etc.

Classic FTA is also not an ideal tool to conduct some system risk analysis which is sensitive to time, due to its ignorance on this factor. Thus, a dynamic fault tree (DFT) is invented to expand the classic FTA with time requirements. Due to its high demand on the capacity of computers to perform time dependent calculations, DFT is not widely applied until the 21st century (Čepin & Mavko, 2001).

2.1.2. Event Tree Analysis

Differing from FTA, event tree analysis starts from the initiating event, building a horizontal inductive tree-like structure from left to right (Andrews & Dunnett, 2000). Branch and branch points mostly stand for operating pathways and dichotomous conditions of the initiating event until the end event occurs (Ferdous et al., 2011).

From the sample system in Figure 2.1, we can build an event tree as follows (Figure 2.3):

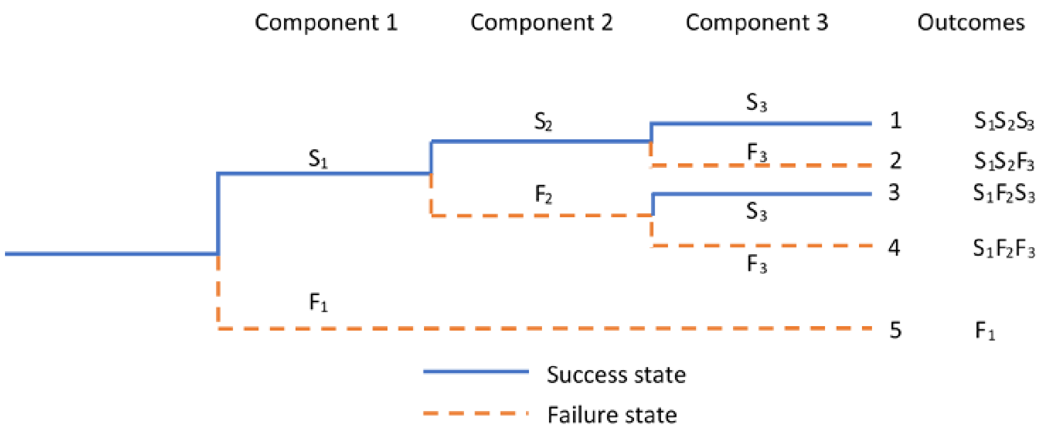


Figure 2.3 ETA Corresponds to the Sample System.

Assume S_i $(i = 1,2,3)$ represents the successful operation probability of component i and the failure of components are F_i $(i = 1,2,3)$. Outcomes 1 to 3 indicate the system working properly; in outcomes 4 and 5, the system is down. Therefore, we can easily obtain the probability of system up and system down:

$$P(System\ Up) = P(Outcome\ 1) + P(Outcome\ 2) + P(Outcome\ 3)$$
$$= S_1S_2S_3 + S_1S_2F_3 + S_1F_2S_3 \quad (2.3)$$

$$P(System\ Down) = P(Outcome\ 4) + P(Outcome\ 5) = S_1F_2F_3 + F_1 \quad (2.4)$$

ETA has the advantages in quickly assessing and identifying potential component failures, inefficient mitigations, and system vulnerability areas in a failures and faults co-existing system (Clemens & Simmons, 1998). Starting from serving the nuclear industry, ETA has gained fame for applications in chemical process, transportation, and offshore oil and gas production (Andrews & Dunnett, 2000). Nevertheless, limitations of this method are also recognized in the practicality of an initiating event and operating pathways (Clemens & Simmons, 1998). For ETA, only one initiating event is allowed, which may be against some situations in a complex system. Moreover, this event should be foreseen by the engineer; otherwise, the subsequent analysis can not be initiated. The specific structure, logic, and loss of each pathway should be known. It indicates that large amounts of data associated with this information should be collected, which is challenging for the project.

2.2. Fuzzy Logic

The concept of fuzzy logic is introduced by Zadeh (1965) to describe the partial truth, where the truth value varies from 0 to 1. By contrast, in Boolean logic, only crisp values, i.e. 0 and 1, are allowed. It is another perspective to explain uncertainty rather than probability theory. Probability theory is based on crisp data to summarize the randomness of occurrence for events. However, in many real-life conditions, it is inappropriate to use crisp data in modeling (Chen et al., 2006), due to the vagueness of information related to many problems. A variety of engineering applications may not identify events in numerical values but linguistic values. For instance, in Figure 2.1, we can define Component 1 as “old”, and the value “old” is a linguistic value. Fuzzy logic enables vague and linguistic information to be quantified into mathematical modeling through fuzzy sets and fuzzy membership functions (Mendel, 1995). A fuzzy set permits a membership function that provides “a measure of the degree of similarity” of an element to the fuzzy sub-sets (Mendel, 1995). However, a crisp set only allows a zero-one membership of an element to its sub-sets. By this idea, fuzzy logic can be seen as a generalization of the traditional uncertainty theory.

When faced with uncertainty, set membership is one of the keys to decision-making (Ross, 2009). The objects in crisp sets precisely satisfy properties of membership. However, imprecise properties of membership are fulfilled by fuzzy sets. We assume an element x in the universe X is either a member of the crisp set A or not. This binary membership can be denoted by an indicator function (Ross, 2009):

$$X_A(x) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases}, \tag{2.5}$$

where $X_A(x)$ represents an indication for the unambiguous membership of element x in set A. If $X_A(x) = 1$, it corresponds that element x has full membership in the crisp set A; if $X_A(x) = 0$, it indicates that x has no membership in A. In fuzzy set theory, the membership of an element is extended to a continuous interval [0,1], where 0 and 1 represent no or full membership for the element. The degree of membership of the element x in a fuzzy set $\underline{A}$ is denoted by:

$$\mu_{\underline{A}}(x) \in [0,1]. \tag{2.6}$$

When the universe of discourse X is discrete and finite, the fuzzy set $\underline{A}$ can be represented as:

$$\underline{A} = \left\{\frac{\mu_{\underline{A}}(x_1)}{x_1} + \frac{\mu_{\underline{A}}(x_2)}{x_2} + \cdots\right\} = \left\{\sum_i \frac{\mu_{\underline{A}}(x_i)}{x_i}\right\}. \tag{2.7}$$

When the universe X is continuous and infinite:

$$\underline{A} = \left\{\int \frac{\mu_{\underline{A}}(x)}{x}\right\}. \tag{2.8}$$

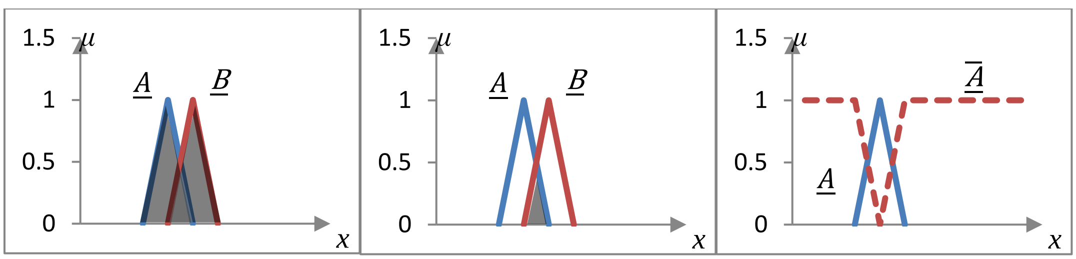
Fuzzy sets follow the operations of set theory, such as union, intersection and complement. We assume fuzzy sets $\underline{A}$ and $\underline{B}$ on the universe X, then operations for fuzzy sets are:

Union $\mu_{\underline{A}\cup\underline{B}}(x) = \mu_{\underline{A}}(x) \vee \mu_{B}(x);$ (2.9)

Intersection $\mu_{\underline{A}\cap\underline{B}}(x) = \mu_{\underline{A}}(x) \wedge \mu_{B}(x);$ (2.10)

Complement $\mu_{\underline{\bar{A}}}(x) = 1 - \mu_{\underline{A}}(x).$ (2.11)

These operations are illustrated in Venn diagrams in Figure 2.4. The shadowed areas represent the actual inclusion of the sets.



(a) (b) (c)

Figure 2.4 Fuzzy Set Operations Illustrated in Venn Diagrams:(a) Union; (b) Intersection; (c) Complement.

Fuzzy logic makes outstanding contributions to the control system field. Fuzzy control breaks through the bottleneck of conventional methods, which is limited by quantitative data insufficiency due to the input-output relations (Lee, 1990). The fuzzy logic controller provides an automatic control strategy that is closer to human thinking and linguistic expressions, and can convert them into the control system (Mamdani, 1977). Numerous control systems, such as a slip power recovery system (Tang & Xu, 1994), a traffic control system (Pappis & Mamdani, 1977), and vehicle suspension systems (Rao & Prahlad, 1997), updated their traditional system with a fuzzy logic controller and resulted in enhanced performance. According to Mendel (1995), fuzzy logic also has applications in scheduling optimization and signal tuning and interpretation.

In a supply chain, the theory of fuzzy logic and fuzzy sets are also applied for their advantages in handling the circumstances such as imprecision preferences, multiple qualitative criteria and incomplete data (Chakraborty et al., 2015). Chen et al. (2006) and Amid et al. (2005) both introduce a fuzzy approach for the supplier selection problem by using linguistic values in rating and weighting quantitative and qualitative factors. Maiti & Maiti (2006) build a multi-objective optimization problem with vagueness in inventory data. Lin et al. (2006) suggest an agility evaluation model assessing the responsiveness of a supply chain to the fluctuating market environment.

Another usage of fuzzy logic is acting as the complement to other risk and uncertainty models. We have discussed the combination of FTA and fuzzy set in section 2.1.1. For ETA, Ramzali et al. (2015) develop a fuzzy method to assist in an offshore drilling system to solve the data shortage problem. By combining ETA, FTA and FMEA (failure mode and effect analysis), Abdelgawad et al. (2011) present a hybrid model for risk analysts to identify critical risk events linguistically in the construction domain. A fuzzy landform classes and datasets of landslides are applied by Gorsevski et al. (2003) in a Bayesian probabilistic model as a decision support tool in forest road planning. De Moraes & Machado (2004) establish a fuzzy hidden Markov model to evaluate an online medical operation simulator. The fuzzy methodology supports the model using continuous variables without information loss. Hassan (2009) utilizes fuzzy logic to obtain the forecast value of a stock price after recognizing data patterns by using a hidden Markov model. Experiments show that this approach has better performance on forecasting accuracy compared to an artificial neural network, an autoregressive integrated moving average, and other hidden Markov-based models.

Although fuzzy logic has expanded our vision on uncertainty to another dimension and it usually generates promising results, it may not be treated as an elixir for any situation. In the work presented by Eierdanz et al. (2008), the difficulty in result validation of fuzzy logic has been found. Another drawback identified by Eierdanz et al. (2008) is the loss of objectivity when defining membership functions and rules of fuzzy logic. Baranyi & Várkonyi-Kóczy (2005) state that when the requirement of modeling accuracy increases, the complexity of fuzzy logic rocketed sharply. Even if complexity reduction techniques are developed, there also are disadvantages in new information adaptation.

2.3. Bayesian Belief Networks

Bayesian Belief Networks (BBNs), according to Pearl (1986), are Directed Acyclic Graphs (DAGs) represented by nodes and directed arcs in between. Nodes are variables in the joint probability distribution factorized by a DAG. The factorization is represented by the directed arcs between the nodes. Conditional probability values represent the quantification of the strengths of these dependencies. A BBN has several interchangeable names, which are Belief Networks, Bayesian Networks and Influence Networks, with distinct focus (Pearl, 1986). The origin of judgments and the nature of quantifiers are emphasized in the former two, while the latter one concentrates on the network directionality. A BBN is an extension of Bayes Theorem that summarizes the mathematical probability propagation that is based on the variables' conditional dependencies (McDonald et al., 2015).

A complete BBN contains two aspects of information. The qualitative aspect is presented as the DAG, and the quantitative aspect is the joint probability distribution that is governed by the DAG structure. The construction of a BBN normally consists of two stages. First, for a specific system, such as a supply chain, a customer service system, or a production line, an identification of the analysis problem for that system should be conducted. With the problem identified, the corresponding variables and the causal relations among them are illustrated by a DAG. Second, the dependency relations that the DAG forms are compiled with the joint probability distribution. These relations are further denoted as a series of conditional probability distributions.

2.3.1. Graph (DAG)

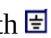
By adapting the theories from Kjaerulff & Madsen (2008), we define $\mathcal{G} = (V, E)$ as a DAG, where V is a set of distinct nodes (or vertices), and $E \subseteq V \times V$ is a set of arcs (or edges). A directed arc from a parent node u to a child node v is denoted as an ordered pair $(u, v) \in E$. The notation $u \rightarrow v$ denotes the relation of $(u, v) \in E$. The parent set and children set of a node v are denoted as $pa(v)$ and $ch(v)$, respectively. A path $\langle v_1, v_2, \ldots, v_n \rangle$ is defined as a sequence of distinct nodes. It is called a directed path if $v_i \rightarrow v_{i+1}$ is valid for each $i = 1,2, \ldots, n-1$. For each $i < j$, v_j is said a descendant of v_i, and v_i is said an ancestor of v_j. The descendant set and ancestor set of a node v are denoted as $de(v)$ and $an(v)$, respectively.

- **Variables**

A set of the mutually exclusive states is represented by a random variable. The set itself also reflects the domain of the variable. The domain of a variable can be either discrete or continuous. A discrete variable has a discrete domain, and a continuous variable has a continuous domain. In this paper, we denote variables or sets of variables as capital letters. Lower case letters are used to denote values of variables. For example, $X = x$ correspond that x is the value of variable X. Besides, the equation can also present that $x = (x_1, x_2, \ldots, x_n)$ is a vector of values for variables $X = (X_1, X_2, \ldots, X_n)$. The domain of X is denoted by $dom(X) = (x_1, x_2, \ldots, x_{\|X\|})$, where $\|X\| = |dom(X)|$ is the number of states of X. In the case $X = (X_1, X_2, \ldots, X_n)$, $dom(X)$ indicates the Cartesian product of the domains of the variables in vector X, i.e. $dom(X) = dom(X_1) \times dom(X_2) \times \ldots \times dom(X_n)$. Particularly,$\|X\| = \prod_i \|X_i\|$, where $i = 1,2, \ldots, n$.

There are two categories of variables. Random variables represent randomly occurring events. Decision variables are the variables under certain choices of some parties, which are typically people. In order to distinguish between variables and nodes (vertices), especially in models that contain utility functions and decision variables, the nodes are denoted as u, v, w, and the sets of nodes are denoted as U, V, W. If a vertex v represents a variable, then the variable is denoted as X_v. If the vertex is a utility function, then the set of random or decision variables are denoted as $X_{pa(v)}$.

- **Evidence**

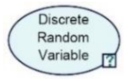
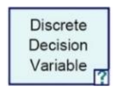
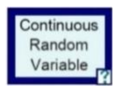
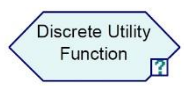
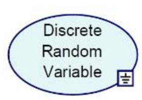
In a probabilistic network, information flows are mainly represented as posterior probabilities in the form $P(x|\varepsilon)$. In this expression, ε is the evidence obtained from an evidence function of a set of variables X, and the evidence function for X is denoted as $\mathcal{E}_X$. Generally, external sources provide information about the evidence. There are two categories of evidence: hard evidence and soft evidence. Hard evidence has an evidence function with zero probabilities to all states but one; otherwise, the evidence is soft evidence. If hard evidence is on a variable X, then X is said to be observed. An observed variable is labeled with  in the DAG.

- **Vertex Symbols**

In this dissertation, ovals are used to indicate discrete random variables, rectangles are used to indicate discrete decision variables, and hexagons are used to indicate discrete utility functions. When the variables and utility functions are continuous, a thick border is used in their symbols.

- **Summary of Notation and Symbols**

Table 2.1 The Summary of Notations and Symbols in BBNs.

Category	Symbol	Notation	Description
Nodes (Vertices)		$U, V, W, \dots$ U_Δ U_Γ $u, v, w, \dots$	sets of nodes the discrete variables sub-set of U the continuous variables sub-set of U nodes
Variables	Discrete Random Variable 	X	variables or sets of variables
	Continuous Random Variable 	X_U	a sub-set of variables that correspond to set of nodes U
	Discrete Decision Variable 	Y_v	variables that correspond to node v
	Continuous Decision Variable 	x_i	states of variables
		x_Y	projection of state x to $dom(Y)$, where $X \cap Y \neq \emptyset$
		$\mathcal{X}$	The set of variables of a network, $\mathcal{X} = X_U$
		$\mathcal{X}_V$	a sub-set of $\mathcal{X}$, where $V \subseteq U$
		$\mathcal{X}_R$	the random variable set of a network
		$\mathcal{X}_D$	the decision variable set of a network
		$\mathcal{X}_\Delta$	the discrete variables sub-set of $\mathcal{X}$
		$\mathcal{X}_\Gamma$	the continuous variables sub-set of $\mathcal{X}$
Utility Functions	Discrete 	$\mathcal{U}$	the utility function set of a network
	Continuous 	$V_\mathcal{U}$	a utility function of the sub-set of V
		$\mathcal{u}(X)$	a utility function with X as domain, where $\mathcal{u} \in \mathcal{U}$
Evidence	For Example, for Discrete Random Variable 	ε	evidence
		$\mathcal{E}_X$	evidence function of X
DAG		$\mathcal{G}$	a directed acyclic graph
Probability		$\mathcal{P}$	a set of conditional probability distributions

2.3.2. Probability

A BBN consists of the DAG and the joint probability distribution. A DAG shows the causal relationships of nodes, and joint probability distribution represents the strengths of the linkages. $P(x)$ denotes the probability of an event x. In the context of a BBN, an event x is a state of a variable X, where $x \in dom(X)$. The basis of treating uncertainty in a BBN is conditional probabilities. The conditional probability of event x, given event y be true, is denoted as $P(x|y)$. There are three axioms that support the probability calculus in a BBN.

Axiom 2.1 $0 \leq P(x) \leq 1$ *for any event* x. *If and only if* x *occurs with certainty,* $P(x) = 1$*; if and only if* x *will not occur with certainty,* $P(x) = 0$.

Axiom 2.2 *For pairwise incompatible events* $x_1, x_2, \dots, x_n$,

$$P(\bigcup_1^n x_i) = P(x_1) + P(x_2) + \cdots + P(x_n) = \sum_1^n P(x_i). \tag{2.12}$$

Axiom 2.3 *The probability that both event* x *and event* y *occur is*

$$P(x\ and\ y) \equiv P(x, y) = P(y|x)P(x) = P(x|y)P(y), \tag{2.13}$$

where $P(x, y)$ *is called the joint probability of the events* x *and* y.

Besides, for a variable X, if the domain $dom(X) = \{x_1, x_2, \dots, x_{\|x\|}\}$, then $\sum_1^{\|x\|} P(x_i) = 1$, and $P(X)$ is a probability distribution, where $P(X) = (P(x_1), P(x_2), \dots, P(x_{\|x\|}))$. If a variable Y is given conditional on a variable X, then the probability distribution is $P(Y|X)$. For any possible state x that is in the domain of variable X, i.e. $x \in dom(X)$, the conditional probability distribution is denoted as $P(Y|x)$. By applying Axiom 2.2, the rule of total probability for the joint probability of two variables X and Y is:

$$P(X) = \left(\sum_{j=1}^{n} P(x_1, y_j), \sum_{j=1}^{n} P(x_2, y_j), \dots, \sum_{j=1}^{n} P(x_m, y_j)\right) = \sum_{j=1}^{n} P(X, y_j), \tag{2.14}$$

where, $dom(X) = \{x_1, x_2, \dots, x_m\}$, $dom(Y) = \{y_1, y_2, \dots, y_n\}$, and $dom(X)$ and $dom(Y)$ are exhaustive sets of mutually exclusive states of variable X and Y, respectively.

For a single variable X and a set of variables $Y = \{Y_1, Y_2, \dots, Y_n\}$, we can denote the conditional probability distributions of probabilistic networks, i.e., BBNs in this context, as $P(X|Y)$. The DAG can be illustrated below in Figure 2.5.

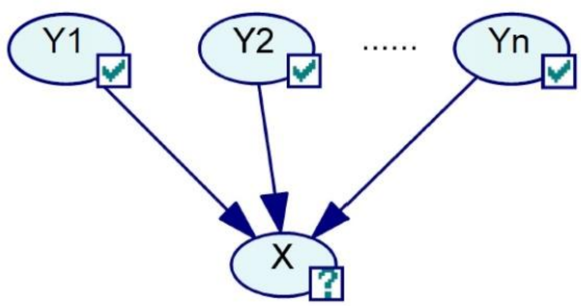


Figure 2.5 DAG Illustration of $P(X|Y)$.

By generalizing Axiom 3.3 to variables X and Y, the following equation can be obtained:

$$P(X, Y) = P(X|Y)P(Y) = P(Y|X)P(X). \tag{2.15}$$

Then, it can be rewritten as Bayes' Rule:

$$P(X|Y) = \frac{P(Y|X)P(Y)}{P(X)}$$

$$= \frac{P(Y|X)P(Y)}{P(X|y_1)P(y_1) + P(X|y_2)P(y_2) + \cdots + P(X|y_{\|Y\|})P(y_{\|Y\|})}. \tag{2.16}$$

For a joint probability distribution with a set of variables $X = \{X_1, X_2, \dots, X_n\}$, the following decomposed product of conditional probability distributions can be obtained repetitively using Axiom 2.3:

$$P(X) = P(X_1|X_2, \dots, X_n)P(X_2, \dots, X_n) =$$

$$P(X_1|X_2, \dots, X_n)P(X_2|X_3, \dots, X_n) \dots P(X_{n-1}|X_n)P(X_n) = \prod_{i=1}^{n} P(X_i|X_{i+1}, X_{i+2}, \dots, X_n). \quad (2.17)$$

A head variable in a conditional probability distribution indicates the variable that is selected to be considered under the conditions of other variables. In equation (2.17), the different patterns of decomposition are determined by the order of selecting the head variables (i.e., X_i's) of the conditional probability distributions. In other words, different patterns of a DAG can be generated by different outcomes of the decomposition. Thus, we can summarize that a specific DAG can only have one joint probability decomposition pattern. Assume that the head variables in a DAG, $\mathcal{G}$, are selected with a topological order $(X_{v_1}, X_{v_2}, \dots, X_{v_n})$, and the parent of node v_i : $pa(v_i) \subseteq \{v_1, v_2, \dots, v_{i-1}\}$ for all $i = 1,2, \dots, n$. Then, the equation above can be simplified as:

$$P(X_V) = \prod_{i=1}^{n} P(X_{v_i}|X_{pa(v_i)}). \quad (2.18)$$

2.3.3. Discrete Bayesian Belief Networks

A discrete Bayesian Belief Network is defined as $\mathcal{N} = (\mathcal{X}, \mathcal{G}, \mathcal{P})$ (Jensen, 2011). The notation shows that the network contains:

- a DAG $\mathcal{G} = (V, E)$ with nodes $V = \{v_1, v_2, , \dots, v_n\}$ and a set of arcs E; and
- a set of discrete random variables $\mathcal{X}$ that is represented by the nodes of $\mathcal{G}$; and

- a set of conditional probability distributions $\mathcal{P}$, and there exists one distribution $P(X_v|X_{pa(v)})$ for each random variable $X_v \in \mathcal{X}$.

Although a discrete BBN is defined with the joint probability distribution and its chain rule decomposed conditional probabilities, the cause-effect relations are often used in the construction of a discrete BBN (Kjaerulff & Madsen, 2008). A graph of nodes, combining with arcs in between can represent the cause-effect relations between entities of a practical problem.

2.3.4. Object-Oriented Bayesian Networks

A complex system is usually composed of collections of similar or identical sub-systems. These sub-systems may have repetitive patterns. A BBN with all of the components in sub-systems visible may cause a chaos for analysts or engineers to construct, maintain or examine the system. In order to enhance the modeling ability of BBNs for systems with such configurations, Object-Oriented Bayesian Networks (OOBNs) are invented to support a large probabilistic model. An OOBN provides a hierarchical view for humans to focus on different levels of abstractions in a complex modeling task.

In an OOBN, we assign an object to each set of variables and related functions, such as probability distributions and utility functions. A set of variables and related functions in a DAG is defined as a network class. An instantiation of a network class within another network class is represented by the object. The object is denoted as M. It is the instantiation of a network class C_M within another network class C_N (Figure 2.6). Assume that three pairwise disjoint sets of nodes $I(C), H(C)$, and $O(C)$ consists a network class C, where $I(C)$ denotes the input nodes, $H(C)$ denotes the hidden nodes and $O(C)$ denotes the output nodes. An object utilizes interface variables, $I(C) \cup O(C)$, to connect to other variables in the same network class. The interface variables indicate the hidden information from the input node to the output node of a network class.

In the OOBN illustration, objects are represented as rounded rectangles. The input variables and output rectangles are represented as dashed or bold ovals, respectively (Figure 2.6).

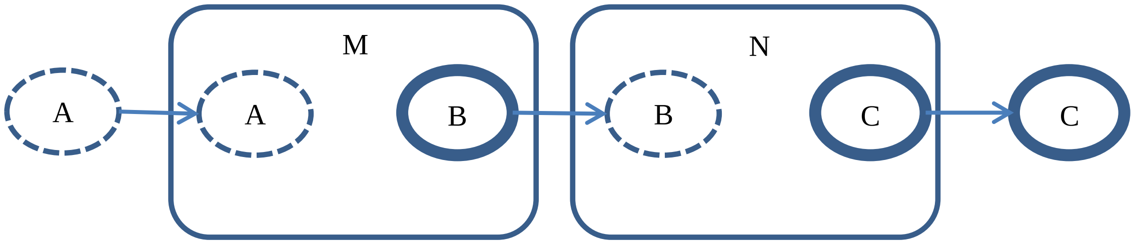


Figure 2.6 An OOBN with Two Network Classes C_M and C_N.

2.3.5. Dynamic Bayesian Networks

BBNs are constrained to be a finite DAG, which does not support time dependent dynamic systems or models with feedback loops. A Dynamic Bayesian Network (DBN) can accomplish those tasks. A BBN can be obtained by unrolling a DBN for the desired number of steps in a multiple feedback loop model. In a time dependent dynamic system, a DBN can be unrolled by a number of time steps to generate a static model. For example, in a dynamic model, a BBN can represent the system at any point in time (Figure 2.7a). In order to monitor the system for a specific period of time n, an analyst builds a DBN for the monitoring system (Figure 2.7b). The variables X_1 and X_3 at time slice t_i have influence on X_1 and X_2 at t_{i+1}, particularly. This category of arcs is defined as temporal arcs. In the compact DBN, a temporal arc has a number in the body of its curve. The number k on the arc indicates that variable at the tail of the arrow at time t_i has influences on another variable at the head of the arrow at time t_{i+k}. Figure 2.7(c) shows an unrolled DBN.

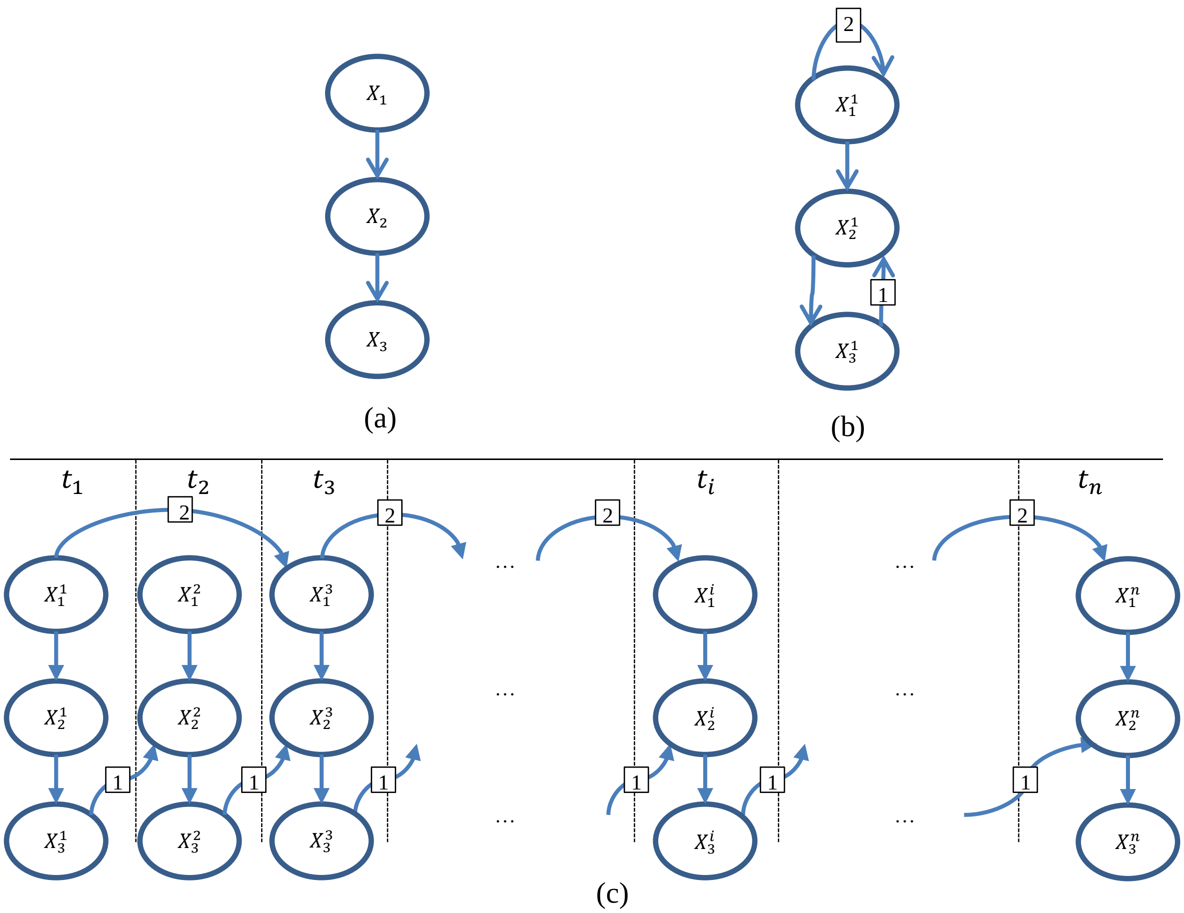


Figure 2.7 (a) A Static BBN Represents the System at any Point of Time;
(b) a Compact DBN; (c) an Unrolled DBN.

2.3.6. Application of Bayesian Belief Networks

Due to the straightforward modeling, a BBN is a proper methodology for systemic risk assessment that shows the cause and effect between nodes (Lee et al., 2009). One of the most applied fields for BBNs is an ecology system risk analysis that is proven to be effective in various environment studies (Dixon & Ellison, 1996; Wolfson et al., 1996; Ticehurst et al., 2007). Marcot et al. (2006) display a causal web of predictor and response variables in an ecological environment by using a three-level BBN model. Marcot et al. (2001) also build BBN models to evaluate fish population viability, and offer suggestions to wildlife conservation. McCann et al. (2006) present that BBNs can be used in ecological research in two ways: one way focuses on the functioning of the whole ecosystem, another emphasizes on the values of each factor.

Project management researchers also show their interests on BBN models for risk analysis. Lee et al. (2009) apply BBN into a large engineering project risk management of a shipbuilding industry. The research identifies the factors in shipbuilding companies that are the most influential. Mitigations to these factors are also stated in order to increase the chance of project success. After determining that over half of software development projects do not reach their objectives, Hui & Liu (2004) construct a BBN model to analyze risks and their impacts in software development projects. The model presents an accurate image of the malfunctioning parts in the project.

Harkleroad et al. (2013) reports on a risk-based method for integrative aviation safety risk modeling and analysis developed and refined by Luxhøj et al. (2003) termed the Aviation System Risk Model (ASRM). Harkleroad et al. (2013) position the ASRM as an exemplar of a risk-based method that is equally influence- and event-based. The ASRM can be used to evaluate the causal factors linked to a hypothesized scenario involving an air vehicle and/or the Next Generation (NextGen) systems and procedures that led to an unsafe state and the interactions among these factors that contributed to the safety risk. The ASRM can also assess the projected impact that new vehicle design changes and/or NextGen systems and procedures may have on potentially reducing the likelihood of significant causal factors. The ASRM uses the flexible, probabilistic approach of Bayesian Belief Networks (BBNs) and influence diagrams to model the complex interactions of aviation system risk factors. The ASRM's systematic approach may be used to guide the construction of an Object-Oriented Bayesian Network (OOBN) with its linkages of numerous sub-networks. Griffin et al. (2015) note that the ASRM is an example of a first generation socio-technical model. The ASRM is described in more detail in Luxhøj (2005). Luxhøj & Sarlo (2012) recently adapt the ASRM method to safety risk modeling for a small UAS.

2.4. System Dynamics

Accelerating growth in economy, society and technology complexity requires decision makers to learn at an increasing rate. In order to ensure the effectiveness in decision-making and learning in a dynamic environment, system thinking is a method to expand the boundaries of thoughts on understanding the behavior of complex systems. System Dynamics (SD) is a modeling tool that provides a system thinking perspective to understand and examine the dynamics of complex systems (Sterman, 2000). SD is an information-feedback approach that studies the interactive influence of organizational structure, time delays and implications to industries (Forrester, 1997). Effective policies and business operations can be designed by using SD through formal computer simulations on the complex systems. The following sections introduce the mathematical formalization of SD modeling as a foundation of the originalDFBN model.

2.4.1. Causal Loop Diagrams

In SD, feedback is one of the core concepts that is represented by a Causal Loop Diagram (CLD). CLDs are ideal tools for capturing the hypotheses on the causes of dynamic flows. The two main elements in a CLD are variables and arrows. By connecting variables, causal links, which are illustrated by arrows, denote the causal relations among the variables. Polarity is assigned to each causal link. By adapting the notations and equations from Sterman (2000), we denote a positive link between variables X and Y as a positive polarity (+), which means the linked variables have the same direction when increased or decreased. The mathematical expression of this relation can be $\partial Y/\partial X > 0$, and in the case of accumulations, $Y = \int_{t_0}^{t} X(\tau)d\tau + Y_{t_0}$, where t_0 is the simulation initial time, t is the simulation termination time, and $X(\tau)$ represents the value of variable X at any time s. A negative link is denoted by a negative polarity (-). The affected variable decreases (increases) if the cause variable increases (decreases) in a negative link. The negative polarity of variables X and Y can be denoted by $\partial Y/\partial X < 0$, and in the case of accumulations $Y =$

$-\int_{t_0}^{t} X(\tau)d\tau + Y_{t_0}$, where t_0 and t correspond the initial time and termination time respectively, and $X(\tau)$ represents the value of variable X at any time s. A loop identifier can be used to highlight the feedbacks in a loop. A positive loop identifier indicates the positive (reinforcing) feedback in the loop; a negative loop identifier indicates a negative (balancing) feedback in the loop.

The sign of the open loop gain method in control theory is used to determine loop polarity. As signal returned by the loop is represented as “gain”. An “open loop” indicates that the calculation of the gain should only be conducted for one feedback cycle. Consider a feedback loop shown in Figure 2.8 that includes variables x_1, x_2, x_3 and x_4. By applying the open loop gain method, the variable x_1 is split into an input variable x_1^I and an output variable x_1^O. The partial derivative of x_1^O with respect to x_1^I is the open loop gain. The partial derivative measures the marginal feedback of x_1 to itself. The sign of the open loop gain, which is the polarity of the loop, is defined as:

$$Polarity\ of\ the\ loop = SGN\left(\frac{\partial x_1^O}{\partial x_1^I}\right), \tag{2.19}$$

where $SGN()$ is the sign function or signum of the partial derivative. $SGN()$ returns $+1$ if the expression inside is positive, and it returns -1 if the expression is negative. If the open loop gain $\partial x_1^O/\partial x_1^I$ is zero, then $SGN(\partial x_1^O/\partial x_1^I) = 0$, which indicates no loops possible for the given variables. The open loop gain can be calculated by the product of the gains of the individual links, i.e. $\partial x_i/\partial x_{i-1}$:

$$SGN\left(\frac{\partial x_1^O}{\partial x_1^I}\right) = SGN\left[\left(\frac{\partial x_2}{\partial x_1^I}\right)\left(\frac{\partial x_3}{\partial x_2}\right)\ldots\left(\frac{\partial x_1^O}{\partial x_n}\right)\right]. \tag{2.20}$$

For the CLD in Figure 2.8, the open loop gain can be calculated by:

$$SGN\left(\frac{\partial x_1^O}{\partial x_1^I}\right) = SGN\left[\left(\frac{\partial x_2}{\partial x_1^I}\right)\left(\frac{\partial x_3}{\partial x_2}\right)\left(\frac{\partial x_4}{\partial x_3}\right)\left(\frac{\partial x_1^O}{\partial x_4}\right)\right]. \quad (2.21)$$

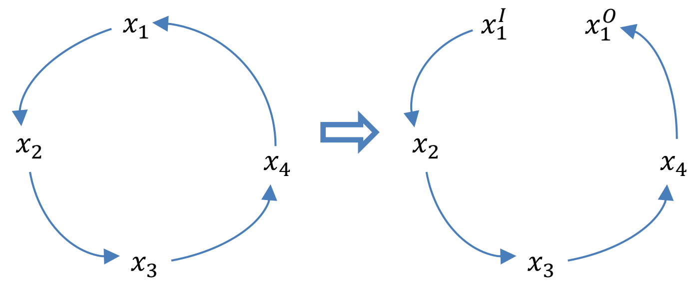


Figure 2.8 Splitting Variable x_1 to Calculate the Polarity of the Loop.

Figure 2.9 shows a personal bank account example CLD. In the example, the bank account balance is determined by both the interest earned and the consumption rate. If the bank account balance is high, then the amount of interest earned increases. The total account balance will be higher if the interest earned is high. Thus, the variables in the left loop of Figure 2.9 boost each other and result in a mutually reinforcing loop. The left loop is called a positive (reinforcing) feedback. For the right loop, bank account balance reinforces the potential rate of consumption. However, the consumption may negatively influence the total amount of the account balance. The influences between variables are balanced in the right loop. Thus, the right loop is called a negative (balance) feedback.

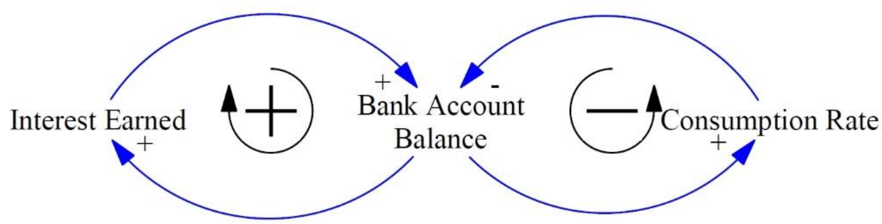


Figure 2.9 Personal Bank Account Example Shows the Notations in a CLD.

A positive or negative polarity does not necessarily represent an actual increase or decrease of the value of a variable. There are two factors that may influence the final result of polarities. First, a variable may be affected by more than one cause variable. In the bank account example, the bank account balance (Y) is influenced by both the interest earned (X_1) and the consumption rate (X_2).

These two influences possess opposite polarities. The increase or decrease of the bank account depends on the strength of the influences, or mathematically, $Y(\tau) = X_1(\tau) - X_2(\tau)$. Second, pure CLDs lack the ability to distinguish between stocks and flows. Stocks are the accumulations of resources in a system, while flows indicate the rates of change in those resources. In the bank account example, the increase in the consumption rate may decrease the total amount in the account balance. However, the decrease in consumption rate may not increase, but decrease the rate of the total balance reduction. Stocks and Flows Illustration (SFI) are introduced to remedy the limitations of a pure CLD.

2.4.2. Stocks and Flows Illustration (SFI)

Stocks are accumulated states of the system that provides the basis for decision-makings. Information about stocks influences decisions in various industries such as purchasing raw materials in manufacturing, changing marketing strategies in retailing, or issuing new debt in banking (Mass, 1980). Stocks enable systems with memory and inertia against changes. Only the change in inflows and outflows can affect the content of a stock. It is not necessary for stocks to be physically tangible. Mental states, such as memories and beliefs, are stocks. Inertia and continuity in people's attitudes and behavior are generated by memories and beliefs. Delays are created by stock reaction lags between their inflows and outflows. During the delays, the accumulated stock of material in the process is the difference between the input and output. The ability to absorb the differences between inputs and outputs reflects practical decision processes. Stocks generate disequilibrium dynamics in systems by decoupling the rates of flows.

Flows are accumulated or integrated by stocks. For a stock and flow system:

$$Stock(t) = \int_{t_0}^{t} [Inflow(s) - Outflow(s)]ds + Stock(t_0), \qquad (2.22)$$

where, t_0 is the simulation initial time, t is the simulation termination time, $Inflow(s)$ and $Outflow(s)$ represent the value of inflow or outflow at any time s (Sterman, 2000).

The rate of change of the stock, i.e. the derivative of the stock with respect to time, equals the net flow into the stock:

$$\frac{d(Stock)}{dt} = Inflow(t) - Outflow(t). \tag{2.23}$$

SFI describe the actual material dynamics in a system. The properly defined units of measure elaborate the variable identities in a complex environment. Stocks are usually measured as a quantity, such as the number of employees, tons of finished products, or dollars spent on consumption. For a specific stock, the correlated flow should be measured in the same unit per time period. For instance, if we measure a product in metric tons, then the rate of production may be denoted by tons per day. The choice of the time period can be unrestrained; however, in order to avoid potential confusion, the time period applied in flows should be universalized in a system simulation.

Figure 2.10 shows a sample to illustrate the diagramming notations for SFI in SD. Stocks (Inventory) are denoted by rectangles indicating containers holding the contents of the stock. Pipes pointing into the stock represent inflows (Purchasing Rate). Pipes pointing out of the stock represent outflows (Selling Rate). The hourglass shaped symbols on the inflows or outflows denote valves controlling the flows. Clouds can both denote the sources and sinks of the flows. The cloud originates an inflow into a stock is a source. The cloud terminates an outflow from a stock represents a sink. It is assumed that sources and sinks possess infinite capacity and flow rate.

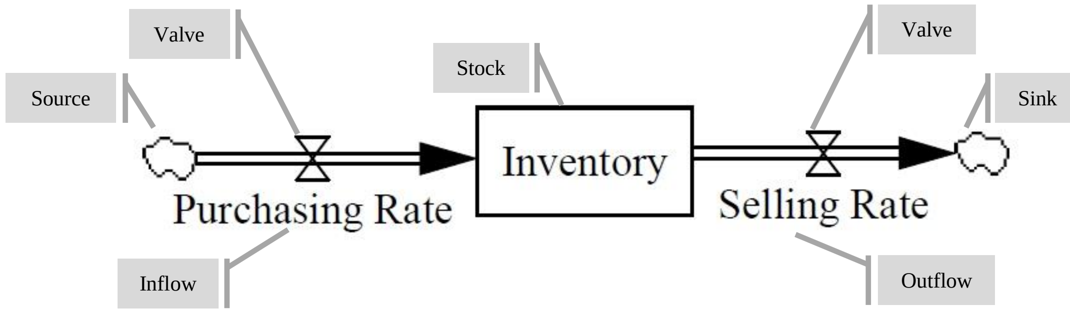


Figure 2.10 A Sample of SFI Model.

In the Figure 2.10 example, a warehouse holds inventory (stocks) for a company that accumulates the inflow of purchasing a specific kind of products and is reduced by the outflow of selling the products. Other inflows and outflows of products are assumed to be zero. The source cloud indicates that there is an unlimited supply of the product. The sink cloud indicates that the demand for the product never grows to a rate that bottlenecks the selling rate.

Auxiliary variables are another category of variables in stock and flow models. An auxiliary variable is a specific stock variable that is the derivative of the variable in a dynamic system that is a non-linear function of exogenous variables, constants, and the stock itself. In matrix form, the rates of stock change $d\boldsymbol{S}/dt$ are represented as a function $f()$ of the stock variables $\boldsymbol{S}$, the exogenous variables $\boldsymbol{U}$ and the constants $\boldsymbol{C}$:

$$\frac{d\boldsymbol{S}}{dt} = f(\boldsymbol{S}, \boldsymbol{U}, \boldsymbol{C}). \tag{2.24}$$

For the example in Figure 2.11, the change rate of the auxiliary variable $Stock1$ can be denoted by the following equation:

$$\frac{dStock1}{dt} = f(Stock1, Stock2, Exogenous\ Variable, Constant). \tag{2.25}$$

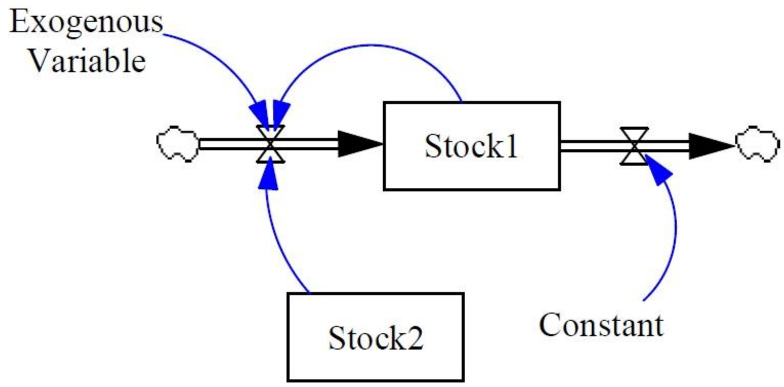


Figure 2.11 A Network of SFI with Auxiliary Variables.

2.4.3. First-Order Linear and Non-Linear Systems

First-order systems are considered to study the behavior of the integrated system of feedbacks with stock and flow structures. In a dynamic system, the number of stocks it contains is defined as orders. It only contains one stock in a first-order system. A linear system or a non-linear system is a system which the rate equation dS/dt is a linear or a non-linear combination of the stock variables and exogenous variables.

2.4.3.1. Linear First-Order Positive Feedback Systems

In a linear dynamic system, the rate equation dS/dt is a weighted sum of the stock variables S_i's and exogenous variables U_j's:

$$\frac{dS}{dt} = Net\ Inflow = (a_1S_1 + a_2S_2 + \cdots + a_nS_n) + (b_1U_1 + b_2U_2 + \cdots + b_nU_n), \qquad (2.26)$$

where a_i and b_j are constants.

For the linear first-order positive feedback dynamic system, the stock variable is denoted by S. The stock variable is the accumulation of its net inflow rate, and the net inflow rate is a linear function of the stock variable. This relation can be mathematically expressed as:

$$S(t) = \int_{t_0}^{t} f(S(\tau))d\tau + S(t_0), \tag{2.27}$$

where

$$f\big(S(\tau)\big) = gS(\tau), \tag{2.28}$$

and g is the fractional growth rate of the stock with units $1/\tau$, t_0 is the simulation initial time, t is the simulation termination time. Figure 2.12 illustrates the structure as a CLD combined with a stock and flow network.

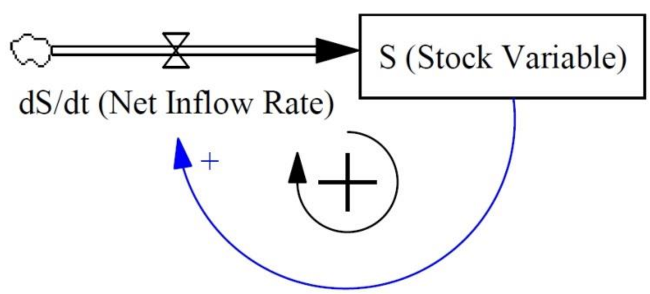


Figure 2.12 A Sample Linear First-Order Positive Feedback System.

By solving Equation 2.28, we can obtain an exponential growth function for the linear first-order positive feedback dynamic system:

$$S(t) = S(t_0)e^{gt}. \tag{2.29}$$

2.4.3.2. Linear First-Order Negative Feedback Systems

Compared with the exponential growth of the positive feedback systems, linear first-order negative feedback systems generate goal-seeking behavior. A general first-order linear negative feedback system contains a stock variable (S), an explicit goal (S^*) and adjustment time (AT). The explicit goal is the desired state of the system. The net inflow is a corrective action that compensates the discrepancy between the desired and actual state of the system. The adjustment time is the expected time to accomplish the corrective action. The net inflow can be denoted by:

$$\frac{dS(t)}{dt} = Net\ Inflow = \frac{f(S(t),S^*)}{AT}. \quad (2.30)$$

For example, in the case illustrated in Figure 2.13, the discrepancy between S and S^* is assumed to be linear, i.e. $f(S(t), S^*) = S^* - S(t)$. The solution of Equation 2.30 is:

$$S(t) = S^* - \big(S^* - S(0)\big)e^{-t/AT}, \quad (2.31)$$

where t_0 is the initial time, t is the current time, and $S^* - S(t_0)$ represents the initial difference between the desired state and the actual state. The current difference between the desired and actual state is denoted by $\big(S^* - S(0)\big)e^{-t/AT}$. As the system time proceeds, the term $\big(S^* - S(0)\big)e^{-t/AT}$ will eventually decay to zero. At that time, the system goal will be reached.

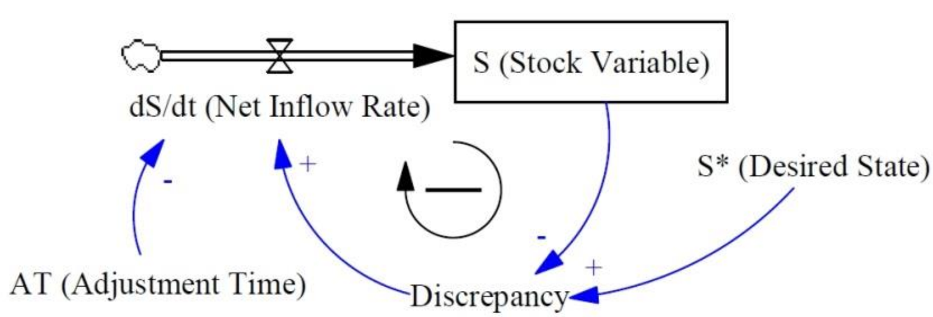


Figure 2.13 A Sample Linear First-Order Negative Feedback System.

In the linear first-order negative feedback system, the stable equilibrium point at $S(t) = S^*$ usually guides the direction of the flow (Figure 2.14). For the system with a lower initial state than the goal, the state of the system increases with a positive net inflow until achieving S^*. For the system with a greater initial state than the desired state, a negative net inflow is formed until S^* is attained. In the example of Figure 2.14, the system goal is determined as 10 units with adjustment time 4 days. The upper curve initiates with $S(0) = 20$, and the lower curve initiates with $S(0) = 0$.

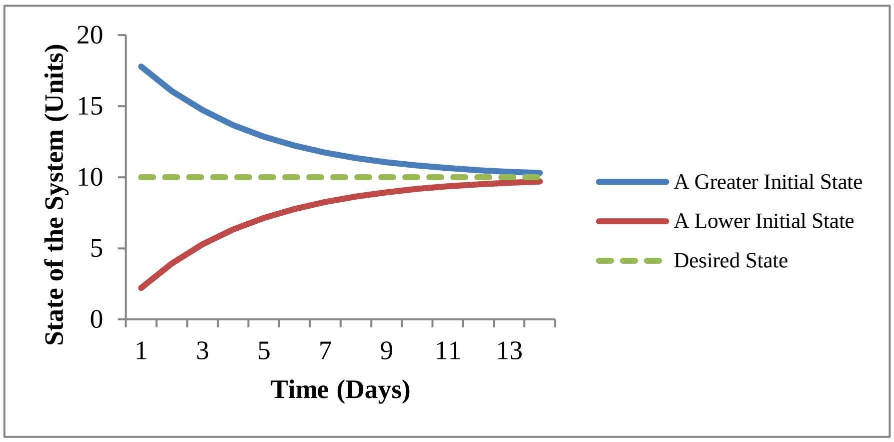


Figure 2.14 The Goal-Seeking Behavior of a Linear First-Order Negative Feedback System.

2.4.3.3. Non-linear First-Order Systems

Systems initially experiencing exponential growth will probably reach the capacity of the environment after a period of time. A non-linear transition from positive feedback domination to negative feedback domination occurs when the limits to growth are approached by the system. The smooth transition from exponential grow to equilibrium state is defined as S-shaped growth.

- **Logistic Growth**

Logistic growth is a special case of S-shaped growth. The original model is published in 1838 by Pierre François Verhulst describing the limits of population growth (Richardson, 1991). In logistic growth, the net fractional population growth rate is a downward sloping linear function of the population:

$$Net\ Birth\ Rate = g(P(t), C)P(t) = g^* \left(1 - \frac{P(t)}{C}\right) P(t), \tag{2.32}$$

where the fractional growth rate $g(P, C)$ is a function of $P(t)$, which is the population at time t, and the capacity C. When the population is close to zero, the fractional growth rate approaches to its maximum g^*. From Equation 2.32, the S-shaped growth can be confirmed: when $P > C$, the net fractional growth rate is negative; when $P = C$, the rate is zero; when $P < C$, the net fractional growth rate is positive. By solving Equation 2.32, an equation about $P(t)$ can be obtained:

$$P(t) = \frac{C}{1+[\frac{C}{P(t_0)}-1]e^{-g^*t}} = \frac{C}{1+e^{-g^*(t-h)}}, \tag{2.33}$$

where $P(h) = 0.5C$. h corresponds the time when the population reaches the half capacity, and it can be solved as:

$$h = \frac{\ln\left[\frac{C}{P(t_0)}-1\right]}{g^*}. \tag{2.34}$$

- **Other Growth Models**

The logistic model is the most commonly used S-shaped growth due to its simplicity and tractability (Sterman, 2000). By relaxing the restriction on the linearity of the fractional growth rate, a number of other S-shaped growth models can be achieved. The Richards curve is a widely-used model (Richards, 1959). The fractional growth rate of the population in Richards curve is non-linear:

$$Net\ Birth\ Rate = \frac{dP(t)}{dt} = \frac{g^*P(t)}{(m-1)}\left[1 - \frac{P(t)}{C}\right]^{m-1}. \tag{2.35}$$

The Richards curve converts to the logistic growth model when $m = 2$. The solution of Equation 2.35 is:

$$P(t) = C(1 - ke^{-g^*t})^{\frac{1}{1-m}}, \tag{2.36}$$

where parameter k is a value that relates to the initial population and capacity.

The Weibull model is another commonly used model in S-shaped growth:

$$P(t) = C\left[1 - e^{-\left(\frac{t}{b}\right)^a}\right], \tag{2.37}$$

where a and b are the shape and scale parameters of Weibull distribution, respectively, and $a, b > 0$. When $a = 2$, it is known as the Rayleigh distribution.

2.4.4. System Dynamics Applications in Supply Chains

Organizations face impediments to learn the behavior of complex systems (Sterman, 2000). System dynamics (SD) has been applied to problems for decades from business practices to medical applications, from military operations to welfare development. SD can be used for a myriad of time-sensitive complex systems with stocks and flows. As a production-distribution system with dynamic complex flows of information, materials, orders, money, people and capital equipment, a supply chain is an ideal problem domain for SD application (Angerhofer & Angelides, 2000).

In the supply chain field, SD is also widely used to understand the complex relationships and flows in the network arrangement, inventory policies, supplier selection, outsourcing, procurement and distribution strategies. Georgiadis et al. (2005) propose an application of SD for food supply chains.

The study uses a CLD to present guidelines for the development of a multi-echelon supply chain. The long-term capacity planning policies of a fast food supply chain with market constraints and transient flows is focused. Demand amplification research is another typical application of SD in supply chains. Anderson et al. (2000) use the machine tool industry to explore the significance of demand amplification on supply chain lead-time, workforce, productivity and inventory. In this study, the large variances in the demand for capital equipment are discovered through the systemic insights on the influences of the slight fluctuations in end-product demand. SD can not only support industrial or academic studies on investigating existing supply chains, but is able to facilitate system establishment in the early stage as well. Vos & Akkermans (1996) apply SD to develop “ex ante” models to support supply chain construction. The models use SD to generate dynamic features to Vos’ (1997) original static model. A supply chain geographical expansion problem is discussed in the study. It generates sensitivity analysis to examine the importance of the processes in the proposed system. The approaches provide valuable insights for supply chain practitioners to migrate their existing systems to new markets with a novel dynamic perspective. For a more globalized and turbulent business environment, Cakravastia & Diawati (1996) presents a supply chain re-design approach by using SD to tackle the challenges. The study investigates the shipbuilding industry in Indonesia. The temporal physical flow, information flow and financial flows are identified with key performance indicators. The model analyzes the behavior of the flows and provides predictive results for each re-design attempt.

In this chapter, five representative risk modeling and uncertainty methods are introduced: Fault Tree Analysis (FTA), Event Tree Analysis (ETA), Fuzzy Logic, Bayesian Belief Networks (BBNs) and System Dynamics (SD). A summarized comparison is presented in Table 2.2. FTA, ETA and Fuzzy Logic are provided as a comparison to the newly developed risk model (DFBN). Principles of BBNs and SD are presented as the mathematical foundation of establishing the DFBN models in Chapter 4.

Table 2.2 Summary of Representative Methods

Existing Supply Chain Risk Analysis Methods	Benefits	Limitations
Qualitative Methods	– Use straightforward risk identification, classification and qualitative result generation – Convenient for decision communication between practitioners – Needs few data as inputs	– Reversed rankings: linguistic outputs present vague image of the outcome of the business (Cox et al., 2005) – Uninformative ratings: depending on different person, the comprehension of linguistic results may differ (Cox et al., 2005)
Quantitative Methods		
• Fault Trees and Event Trees	– Use straightforward AND/OR logic – Can assess and identify component failures, inefficient mitigations, and system vulnerability areas quickly (Clemens & Simmons, 1998)	– Model becomes clumsy when presenting complicated relationships between events – Hard to collect failure data – Hard to handle large scale computation – May generate unreliable results with inaccurate input data (Khan & Abbasi, 2000) – Only one initiating event is allowed – The events should be foreseen by the engineer, or the subsequent analysis can not be commenced
• Fuzzy Logic	– Quantifies the linguistic information into mathematical modeling (Mendel, 1995) – Overcomes the data insufficiency problem in quantitative modeling (Lee, 1990) – Capable of handling imprecision preferences, multiple qualitative criteria and incomplete data (Chakraborty et al., 2015)	– Difficult to validate results (Eierdanz et al., 2008) – Easy to lose objectivity in membership function definitions (Eierdanz et al., 2008) – Complexity of model rockets sharply when the requirement of modeling accuracy increases (Baranyi & Várkonyi-Kóczy, 2005) – Even if complexity reduction techniques are developed, new information is still hard to adapt (Baranyi & Várkonyi-Kóczy, 2005)
• Bayesian Belief Networks	– Straightforward arc and node modeling for complex and large scale systemic risk assessment (Lee et al., 2009) – Contain both qualitative and quantitative assessment of events – Flexible probabilistic approach, easy to adapt to various application scenarios (Harkleroad et al., 2013)	– Hard to collect conditional probability data (Uusitalo, 2007) – Lack the ability to deal with continuous data (Uusitalo, 2007) – Acyclic networks, information feedbacks are not supported (Nielsen & Jensen, 2009)
• System Dynamics	– Enables the dynamic feedback flows in analyzing the interactive influence of organizational structure, time delays and implications to industries (Forrester, 1958) – Convenient for understanding complex relationships and flows in network arrangement, inventory policies, supplier selection, outsourcing, procurement and distribution strategies in a supply chain (Angerhofer & Angelides, 2000)	– Easy to be misapplied (Featherston & Doolan, 2012) – The objectives and expected outcomes of the model may be often misinformed (Featherston & Doolan, 2012)

In the following chapter, the taxonomy of supply chain risk analysis is introduced to provide a modeling framework and an industrial data reference for the DFBN models.

3. Taxonomy of Supply Chain Risk Analysis

This chapter provides insights on a taxonomy of supply chain risk analysis to elaborate the features and capabilities of existing fields of study in supply chain management. The previous chapter focuses on quantitative risk analysis tools. This chapter introduces essential ideas from the perspective of supply chain studies to analyze the foundations of this complex system. The chapter follows a descending taxonomy order, starting from supply chain management to supply chain risk management and quantitative supply chain risk analysis.

3.1 Supply Chain Management

According to Angerhofer & Angelides (2000), a supply chain is a system with dynamic complex flows of information, materials, orders, money, people and capital equipment through the processes of production and distribution. Supply chain profitability is defined as the difference between gaining from the customer and the overall cost across the supply chain. Maximizing profitability is one of the objectives of a supply chain (Beamon, 1998). The decisions in successful supply chain management should be made to ensure profitability.

3.1.1. An Analytical Framework for Supply Chains

With the goal of gaining profit by fulfilling a customer's request, all the business parties that involve manufacturing, supplying, distributing, warehousing and retailing are categorized as the elements of a supply chain (Chopra & Meindl, 2010). A typical supply chain contains five stages: customers, retailers, wholesalers/distributors, manufacturers and raw material suppliers. The flow of goods, information and capital builds the connections between stages. The processes of flows in a supply chain can be thought in two perspectives: a cycle view and a push/pull view. With the cycle view, the processes are discerned as a series of cycles between two successive stages in a supply chain, such as customer order cycle, replenishment cycle, manufacturing cycle and procurement cycle. The push/pull view focuses on the interaction of supply chain activities with customers. Push processes represent the initial activities that anticipate customers' orders. Pull processes, on the contrary, are exercised passively by customers' orders (Ahn & Kaminsky, 2005). In a macro perspective, supply chain processes can be classified into three high-level procedures: customer relationship management (CRM), internal supply chain management (ISCM) and supplier relationship management (SRM).

- Supply Chain Drivers

Drivers determine performance by influencing the flows of supply chains in terms of responsiveness and efficiency (Chopra & Meindl, 2010) (Figure 1.1). There are three logistical drivers: facilities, inventory and transportation; and three cross-functional drivers: information, sourcing and pricing.

  - Facility is one of the logistical drivers that represent the actual physical location where products are stored or manufactured. Significant impact on the performance of the supply chain is determined by the decisions on role, location and capacity of facilities.
  - Inventory is another supply chain logistical driver that considers all raw, work-in-process or finished goods. Inventory policies, including cycle inventory, safety inventory, seasonal inventory and the level of product availability affects the efficiency and responsiveness dramatically.
  - Transportation is the source of the physical flows of goods in a supply chain. The design and operation of transportation networks significantly contribute to the liquidity of the system.
  - Information consists of data and analysis regarding other supply chain drivers. Information directly affects other drivers and can potentially develop into the most effective drivers among others.

- Sourcing is the decision-making process that aims to choose particular parties to accomplish specific supply chain activities. The main decision is focusing on the in-housing or outsourcing designated supply chain tasks.
- Pricing determines the charges for goods and services available in the supply chain. Although the rise in price influences the return of goods or services for each transaction, the total revenue or profit may not increase simultaneously. This phenomenon attributes to the behavioral change of the customer due to different pricing strategies.

After identifying the drivers affecting supply chain performance, various plans can be applied to different processes in supply chains.

3.1.2. Demand Management

- Demand Forecasting

Estimation of future demand constitutes one of the most important tasks for supply chain decision-making (Chopra & Meindl, 2010). From the push/pull view of the supply chain, both the push and pull processes either anticipates or responds to customers' demand. Thus, supply chain planning requires reliable demand forecast results. In order to generate accurate demand forecast, a company should be well-informed with the related factors to the forecast, such as past demand, lead time, economy status, pricing and marketing strategies and competition information (Chen et al., 2000). Four types of forecasting models are classified: qualitative methods, time-series methods, causal methods and simulation (Chopra & Meindl, 2010).

- Qualitative methods are based on subjective judgments on activities. The methods are effective when historical data is scarce.

- Causal methods assume that certain factors in the environment are correlated with the demand while correlations are not causations, the causal relationships between demand and environmental factors may be used to forecast future demand.
- Simulation imitates the customer behaviors in supply chain processes. Simulation combines the time-series and causal methods to analyze the impact of decisions and proposals of various parties in the supply chain.
- Time-series methods generate predictive results from historical demand data. It is assumed that the past data act as a proper indicator for demand forecasts. In time-series methods, the historical, or observed demand consists of a systematic component and a random component:

$$Observed\ Demand\ (O) = Systematic\ Component\ (S) + Random\ Component\ (R).$$

The systematic component is the predictable part of the demand, and the random component provides the variation of the forecast. The value of the systematic component is denoted by:

$$Systematic\ Component = (Level + Trend) \times Seasonal\ Factor,$$

where $Level$ is the expected value of demand, $Trend$ is the rate of demand growth or decline for the next period, and $Seasonal\ Factor$ is the predictable seasonal demand fluctuations.

- Aggregate Planning

Aggregate planning can be performed based on the demand forecast to make decisions on production, inventory, backlogs and outsourcing in a supply chain with limited production capacity and lead times. Aggregate planning uses a macro view on problem-solving rather than a stock-keeping unit (SKU) level method. After identifying operational parameters over a specific time horizon, such as production rate, workforce, overtime, machine capacity, subcontracting, backlog and inventory on hand, aggregate planning can be performed to satisfy the forecast demand with maximized profit (Singhvi & Shenoy, 2002). Linear programming can be applied to solve an aggregate planning problem with a maximizing total profit objective function and constraints on production.

3.1.3. Inventory Management

The studies of inventory management mainly concentrate on cycle inventory, safety inventory and product availability.

- Cycle Inventory

The average amount of inventory that satisfies demand is cycle inventory (Chopra & Meindl, 2010). As a result of the production, procurement and transportation in large lots, cycle inventory are used to exploit economies of scale in supply chains. If we assume a steady demand, the cycle inventory is can be denoted by:

$$Cycle\ Inventory = \frac{Q}{2}, \tag{3.1}$$

where Q is the lot size. In practice, estimation of the ordering and holding cost supports the decision on cycle inventory level. The holding cost can be obtained from the sum of the cost of capital, obsolescence costs, handling costs, occupancy costs and miscellaneous costs. According to Marcus (2013), the weighted-average cost of capital (WACC) is a common method that used to measure the amount of cost of capital, and it can be denoted as:

$$WACC = \frac{E}{D+E}\left(R_f + \beta \times MRP\right) + \frac{D}{D+E} R_b (1-t), \tag{3.2}$$

where E is the amount of equity, D is the amount of debt, R_f is the risk-free rate of return, β is the beta of the company, MRP is the market risk premium, R_b is the rate of borrowing money and t is the tax rate. In addition, $R_f + \beta \times MRP$ can be seen as the cost of equity of the company. For ordering cost, it consists of buyer time costs, transportation costs and receiving costs.

In order to achieve optimal lot size of a product, the economic order quantity (EOQ) technique is applied:

$$EOQ = \sqrt{\frac{2DS}{hC}}, \tag{3.3}$$

where D is the annual demand of the product, S is the fixed cost per order, C is the unit cost of the product and h is the holding cost per year per unit of product. From Equation 3.3, the optimal ordering frequency n^* can be obtained:

$$n^* = \frac{D}{EOQ} = \sqrt{\frac{DhC}{2S}}. \tag{3.4}$$

- Safety Inventory

In a supply chain, safety inventory is carried to prevent product shortage from the possible demand exceeding situation that was not accurately forecasted (Chopra & Meindl, 2010). The necessity of safety inventory attributes to the uncertainty of demand. If we assume that the demand is normally distributed with mean D_i and variance σ_i^2 for a period $1,2, \ldots, L$, then the total demand during lead time L is also a normal distribution with mean $P = D_L = \sum_1^L D_i$ and variance $\Omega^2 = \sum_1^k \sigma_i^2 + 2\sum_{i>j} \rho_{ij}\sigma_i\sigma_j$, where ρ_{ij} is the correlation coefficient of demand between periods i and j.

The expected shortage per replenishment cycle (ESC) measures the average unfulfilled amount of demand per replenishment cycle. Assume that the demand distribution during the lead time is denoted by $f(x)$, and ROP corresponds to the reorder point, then the ESC is given by:

$$ESC = \int_{ROP}^{\infty} (x - ROP) f(x) dx. \tag{3.5}$$

If the demand is normally distributed with mean D_L and variance σ_L^2 during the lead time L, given a safety stock level ss, then ESC can be denoted by:

$$ESC = -ss\left[1 - F_s\left(\frac{ss}{\sigma_L}\right)\right] + \sigma_L f_s\left(\frac{ss}{\sigma_L}\right), \tag{3.6}$$

where F_s and f_s are the standard normal cumulative and density function, respectively.

- Optimal Product Availability

The level of product availability is critical for the responsiveness of a supply chain (Chopra & Meindl, 2010). A high level of availability yields a high customer demand satisfaction and increased revenue. However, the trade-off of a high level of availability is the requirement of a high inventory level, which may probably increase the holding cost and the risk of overstocking. The cycle service level is utilized to measure the level of product availability. The optimal cycle service level CSL^* can be achieved by the following equation:

$$CSL^* = \frac{C_u}{C_u + C_o}, \tag{3.7}$$

where C_u is the cost of understocking by one unit, and $C_o$is the cost of overstocking by one unit. If the demand during a season follows a normal distribution with mean μ and variance σ^2, then the optimal order quantity O^* can be denoted by:

$$O^* = F^{-1}(CSL^*, \mu, \sigma). \tag{3.8}$$

3.1.4. Transportation Networks

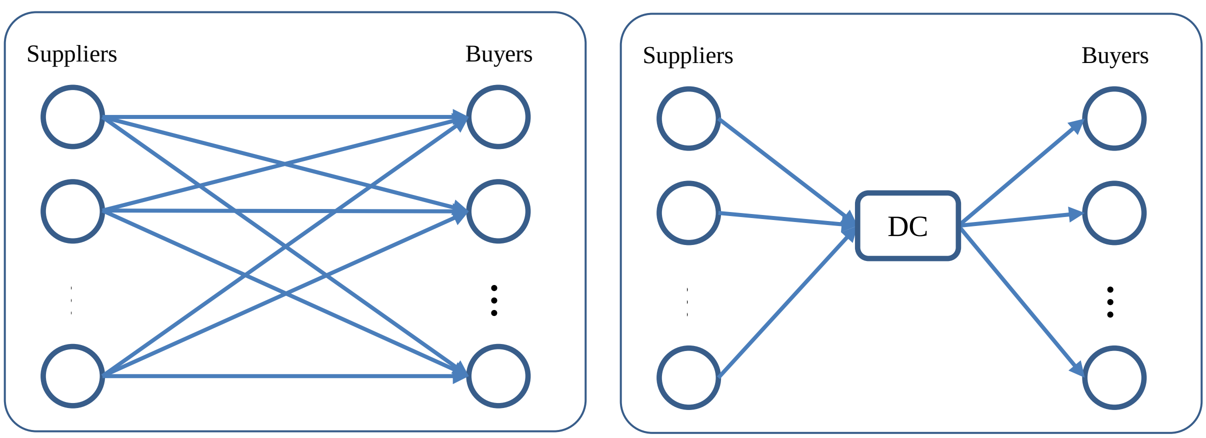
The performance of a supply chain is affected by the design of a transformation network. The influence on the performance is reflected in operational transportation decisions about infrastructure establishment, scheduling and routing (Chopra & Meindl, 2010). The decisions on transportation also include the choice between different transportation modes, such as air, package carriers, truck, rail, water, pipelines etc. The design of a transportation network can be mainly categorized into two methods: direct shipping and shipping via distribution centers.

3.1.4.1 Direct Shipping

Buyers directly receive all shipments from suppliers in direct shipping (Figure 3.1a). Shipment routing is specified in a direct shipment network. The decisions on this shipping option focus on shipment volume, transportation mode and shipment schedule. Direct shipping can eliminate the process in the intermediate warehouse. Other proportions in the shipping network are not affected by the decisions made for direct shipping. This method shortens the delivery time and simplifies the operation and coordination.

3.1.4.2 Shipping via Distribution Centers (DCs)

A DC is built for each geographic region and suppliers firstly send shipments to DCs instead of buyers under this option. Then, shipments are divided by DCs and forwarded to buyers' location (Figure 3.1b). DCs play two roles in a transportation network: inventory storages and transfer hubs. The advantage of the DC contains reducing supply chain costs by achieving economies of scale for inbound transportation. DCs allow suppliers, which are located far from the buyers, to send large shipments, and then DCs can deliver the divided small shipments to nearby buyers at a low cost.



(a) (b)

Figure 3.1 The Design of Transportation Networks: (a) Direct Shipping; and (b) Shipping via DCs. (Source: Adapted from Chopra & Meindl, 2010)

3.1.4.3 Trade-offs in Transportation Design

In a supply chain network, shippers should consider the trade-offs of all transportation decisions on facility costs, processing costs, inventory costs and the customer responsiveness (Chopra & Meindl, 2010).

- Transportation and Inventory Cost

According to Chopra & Meindl (2010), transportation and inventory cost trade-offs are affected by the decisions on the transportation mode and inventory aggregation. For transportation mode decisions, transportation and inventory costs should be balanced. For example, in a supply chain, a lower-cost transportation mode does not necessarily result in a lower total cost due to possible higher holding costs or delay penalties. In aggregate planning, companies can reduce the safety inventory. This technique is widely applied by on-line shopping sites to lower their holding cost. However, the trade-off is the increase in transportation cost.

- Transportation Cost and Customer Responsiveness

High responsiveness transportation enables customers receiving their orders in a short time with a high transportation cost. Economies of scale with a lower transportation cost occur when the customer responsiveness decreases because larger shipment volume can be achieved in each shipment.

In summary, the total cost and service quality can be affected by all of the decisions made. Thus, it is important for a company to consider a most suitable transportation design to fulfill both the customer's and company's need in a specific business environment.

3.1.5. Cross-Functional Drivers

Two cross-functional drivers, pricing and information technology, are introduced to present the interactions of traditional drivers (facilities, inventory and transportation) from a novel perspective.

3.1.5.1 Pricing and Revenue Management in a Supply Chain

In order to match supply and demand in a market better, pricing is an important tool to enhance supply chain profits. Revenue management utilizes pricing to maximize profits from limited supply chain assets, which includes capacity and inventory (Chopra & Meindl, 2010). Revenue management makes remarkable contributions to supply chain profitability when a product's value differs from distinct market segments. For example, according to the supply and demand theory in microeconomics, we assume that the demand curve for a normal good is shown in Figure 3.2. The whole triangle area bounded by the demand curve is the maximum revenue that a company can obtain from selling this good. However, if a single price $6 is set, only a partial of revenue (shadowed area) can be obtained. By applying revenue management techniques, multiple prices for different customer segments can be formulated and the total revenue is increased.

In the practical situation, distinct customer segments may have different demand curves. According to (Chopra & Meindl, 2010), we assume that the linear demand curve d_i for customer segment i is denoted as:

$$d_i = A_i - B_i p_i, \quad (3.9)$$

where p_i is the price decided for segment i, A_i is the quantity demand when the price drops to 0 for segment i, and B_i is the slope of the curve for segment i.

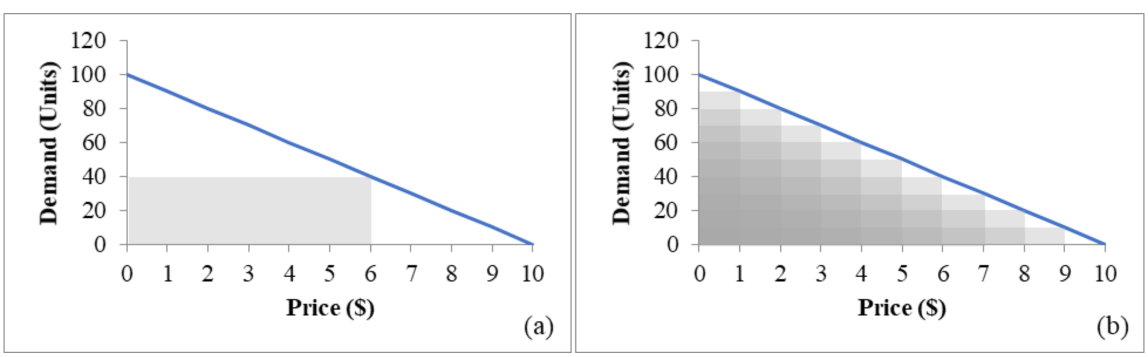


Figure 3.2 The Revenue Obtained from: (a) Single Pricing; and (b) Multiple Pricing.

For a total k distinct customer segments, the profit can be maximized by:

$$Max \sum_{i=1}^{k} (p_i - c)(A_i - B_i p_i), \tag{3.10}$$

where c is the cost of production per unit for the product.

For segment i, if the capacity is unlimited, the problem is maximizing:

$$(p_i - c)(A_i - B_i p_i). \tag{3.11}$$

Thus, the optimal price for segment i can be denoted as:

$$p_i = \frac{A_i}{2B_i} + \frac{c}{2}. \tag{3.12}$$

If the capacity of the supply chain for the product is Q, then:

$$\sum_{i=1}^{k} (A_i - B_i p_i) \leq Q, \tag{3.13}$$

$$A_i - B_i p_i \geq 0 \quad for\ i = 1,2, \dots, k. \tag{3.14}$$

3.1.6. Value-Based Supply Chain Management

After obtaining the supply chain drivers, such as demand forecast, inventory level, transportation and pricing strategies, Brandenburg (2013) suggests the value-based supply chain management model to measure shareholder value of a company and economic value. The quantitative value-based model is based on discounted cash flow. By using published data, the value-based model allows practitioners estimating the impacts of shareholder value from supply chain drivers.

In the value-based model, supply chain cost consists of the cost of goods sold (COGS) and logistics cost:

$$Supply\ Chain\ Cost = COGS + Logistics\ Cost. \tag{3.15}$$

COGS reflects the physical production costs and sale costs of products (Poston & Grabinski, 2001). Transportation cost, distribution cost, inventory carrying and administration cost are the components of logistics cost (Shoshanah & Roussel, 2005). Lee (2008), Winkler et al. (2006) and Ritchie & Brindley (2007) also state that supply chain cost may be highly relevant to supply chain cost.

Working capital is one of the metrics measuring the economic value and the liquidity position of a company (Schilling, 1996). Working capital equals the sum of inventory and trade receivables with trade payables deducted (Brealey et al., 2008):

$$Working\ Capital = Inventory + Trade\ Receivables - Trade\ Payables. \quad (3.16)$$

3.2. Supply Chain Risk Management

A supply chain is vulnerable to various risks, especially for the globalized supply chains nowadays. Any single inharmonious occurrence in a global supply chain network may cause disruptions in another part of the world. The risks, according to Chopra & Meindl (2010), include supply delays, supply disruption, price fluctuation, demand fluctuation and exchange-rate fluctuations etc. The detailed risk drivers are listed in Table 3.1.

The design of a supply chain network should possess a high capacity and flexibility in implementing mitigation plans. However, a mitigation plan on a risk driver may be a trade-off to another driver. For instance, in order to mitigate the risk of stock out, a larger amount of inventory can be purchased. As a trade-off, the risk of holding cost will increase. Thus, the rational use of mitigation plans should also be considered by supply chain managers. Table 3.2 provides a list of available mitigation plans and tailored strategies.

Table 3.1 Supply Chain Risks and Drivers.
(Source: Adapted from Chopra & Meindl, 2010)

Category	Risk Drivers
Disruptions	Natural disaster, political variation, strike or supplier bankruptcy.
Delays	Supplier high capacity utilization or inflexibility.
Systems Risk	Information interruption, system updating.
Forecast Risk	Forecast inaccuracy due to seasonality, product variety, long lead times, short life cycles, small customer base or information distortion.
Intellectual Property Risk	Vertical industry integration, outsourcing.
Procurement Risk	Exchange rate change, raw-material price, industrial capacity overload, procurement from single source.
Receivables Risk	Number of customers, financial strength of customers.
Inventory Risk	Product obsolescence, inventory holding cost, depreciation, demand and supply uncertainty.
Capacity Risk	Capacity inflexibility, capacity cost.

Table 3.2 Available Risk Mitigation Plans and Tailored Strategies.
(Source: Adapted from Chopra & Meindl, 2010)

Risk Mitigation Plans	Tailored Strategies
Increase Capacity	For predictable demand: decentralize capacity. For unpredictable demand: centralize capacity. Increase decentralization when the cost of capacity drops.
Get Redundant Suppliers	For high-volume products: get more redundancy For low-volume products: less and centralized redundancy.
Increase Responsiveness	For commodity products: prefer cost to responsiveness. For short-life-cycle products: prefer responsiveness to cost.
Increase Inventory	For predictable or low-value products: decentralize inventory. For unpredictable or high-value products: centralize inventory.
Increase Flexibility	For predictable or high-volume products: prefer cost to flexibility. For unpredictable or low-volume products: prefer flexibility to cost. For high-cost products: centralize flexibility.
Pool or Aggregate Demand	As unpredictability grows: increase aggregation.
Increase Source Capability	For high-value or high-risk products: prefer capability to cost. For low-value or commodity products: prefer cost to capability. Centralize high capability in flexible sources.

3.3. Supply Chain Risk Taxonomy

A supply chain is a complex system with systemic dynamic flows between the six supply chain drivers (logistical drivers: facilities, inventory and transportation; and cross-functional drivers: information, sourcing and pricing). During supply chain operations, risks conceal in the drivers and are ready to disrupt the system with any related trigger. In order to facilitate supply chain practitioners identifying and mitigating supply chain risks, the Supply Chain Risk Identification and Prevention Taxonomy (SCRIPT) is introduced in this study. By using the six drivers as the framework of this taxonomy, SCRIPT is established and shown in Figure 3.3.

SCRIPT is a convenient reference for supply chain researchers and practitioners establishing risk analysis tools. For both qualitative and quantitative models, SCRIPT can be used as a guidance of extracting risk factors from events. For instance, in a supply chain, the user can firstly identify the risk is either internal or external to the company. Then, a category of risk factors can be determined, and the cause of the event can be selected from the SCRIPT for further modeling. If the prospective model includes a BBN, then the chosen risk factor can be one of the nodes in the BBN.

Chopra & Sodhi (2004) present fundamentals of the supply chain risks and their drivers of SCRIPT. However, SCRIPT categories supply chain risks from each company's business environment. According to Ravindran & Warsing (2012), the risks in a supply chain are from the external and internal environment of a company. SCRIPT follows this taxonomy and further divides external risk factors into natural-related and human-society-related environments. Due to the complexity of each supply chain system, complete and exhaustive risk taxonomy seems difficult to demonstrate in a single model. For example, Trucco et al. (2008) state over 70 metrics to measure the systemic risks of a hybrid maritime transportation system of a supply chain. Barkan et al. (2003) analyze 47 hazardous events that affect the normal operation of a railway transportation system. Sumner (2000) and Benaroch et al. (2006) elaborate the information technology risks that companies may encounter. SCRIPT consolidates the risk factors in a general form and makes referable comprehensive risk taxonomy for most of the companies in the supply chain, including supplier, manufacturer, wholesaler/warehouse and retailer.

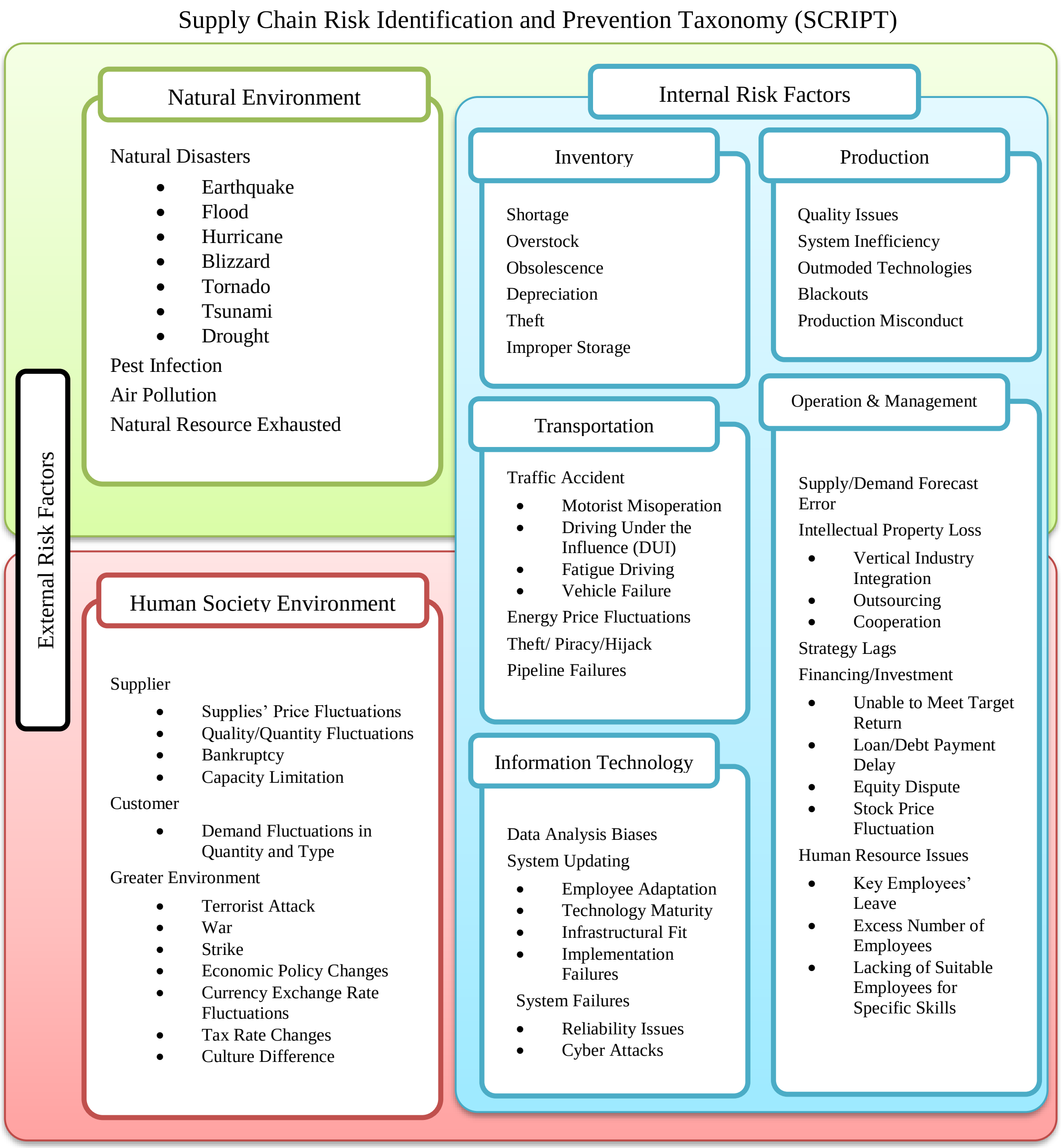


Figure 3.3 Supply Chain Risk Identification and Prevention Taxonomy (SCRIPT).

In this chapter, the taxonomy of supply chain risk management analysis is presented. The elements in a supply chain analysis are introduced through supply chain framework design, demand management, inventory management, transportation networks and cross-functional supply chain drivers. Supply chain risks, drivers, and mitigations are also provided. In the following chapter, by utilizing the concepts presented in Chapter 2 and Chapter 3, a mathematical inference of the DFBN, the modified ODFBN, and an application in the supply chain domain for each of the models are demonstrated.

4. Methodology for Complex Temporal Uncertainty Modeling

This chapter introduces the development of the Dynamic Flow Bayesian Network (DFBN) and the modified Optimized Dynamic Flow Bayesian Network (ODFBN). First, the top tier System Dynamics (SD) model, which illustrates the stock and feedback flows between different business units and parties, can be constructed by referring to the business framework of companies. Then, by utilizing SCRIPT, the DFBN models possess the ability to construct the risk factors and their causal relationships in key business units or parties. Second, this chapter also provides uncertainty analysis techniques to evaluate the systemic risks and their influences on the business performance of the company. Third, a small application of the DFBN is conducted by using a specific supply chain scenario. Finally, the ODFBN is introduced with an application in the same scenario as the DFBN case study.

4.1. Development of Hybrid Temporal Belief Network Method

Existing risk models provide firm analysis results for various industry domains. However, the usage of these results may be limited by the form of outputs (i.e., degree of similarity of fuzzy logic and probability values from most of the probabilistic models). The DFBN provides a hybrid approach that uses the same type of data with the existing risk models and generates more user-friendly outputs for business practitioners. The following sections start from the establishment of a top-level SD model, then move to the identification risk factors and DBNs, and finally to the integration of the DFBN.

4.1.1. Dynamic Flow System (System Dynamics)

A DFBN describes the fluctuations of stocks and flows that are affected by systemic risks. The top-level dynamics of a DFBN is modeled by applying an SD approach. According to Chapter 2, an SD model usually contains Causal Loop Diagrams (CLDs) and the illustration of stocks and flows. Variables in a CLD represent the flow rates of people, money, inventory, information etc. (Sterman, 2000). For a system, a variable

$$Y^t = X_1^t - X_2^t, \tag{4.1}$$

where X_1^t and X_2^t are positive and negative feedback variables for Y at time t, respectively. A pure CLD presents the instantaneous flow rates without the memory of their states in the previous time. Thus, the cumulated number of stocks in a system cannot be recorded. The stock and flow diagram is originated by Forrester (1997) to describe the dynamics of inventory. The essence of the diagram is the accumulated states of the system, i.e., stocks, can be introduced in the SD model. Stocks enable systems with inertia against changes, such as inflows and outflows. Thus, if we assume the system is considered in continuous time, by adapting Equation 2.22 (Sterman, 2000), the stock level S at time t can be denoted as:

$$S^t = \int_{\tau=t_0}^{t} f(I_1(\tau), I_2(\tau), \dots I_i(\tau), \dots, I_n(\tau), O_1(\tau), O_2(\tau), \dots O_j(\tau), \dots, O_m(\tau)) d\tau + S^{t_0}; \text{ or} \tag{4.2}$$

in discrete time:

$$S^t = \sum_{\tau=1}^{t} f(I_1(\tau), I_2(\tau), \dots I_i(\tau), \dots, I_n(\tau), O_1(\tau), O_2(\tau), \dots O_j(\tau), \dots, O_m(\tau)) + S^{t_0}, \tag{4.3}$$

where $I_i(\tau)$ and $O_j(\tau)$ are the inflow and outflow rates respectively, $i \in \{1,2, \dots, n\}$, $j \in \{1,2, \dots, m\}$, n and m are the number of inflow and outflow variables, respectively. t_0 is the start time, and T is the end time. The number of inflows and outflows are n and m, respectively. Equation 4.1 and 4.2 are the basis of the top-level SD model. When applying in the supply chain domain, various stock analysis techniques can be combined with the SD model to generate optimized inventory management plans. In the top-level SD model, stock and flow rate data can be directly obtained from industrial operations.

For example, we assume a dynamic flow system with four variables: Purchasing Rate (F_1), Inventory Level (S), Selling Rate (F_2) and a Feedback Variable (FV) (Figure 4.1). FV is an original term that describes a variable with influence on other variables.

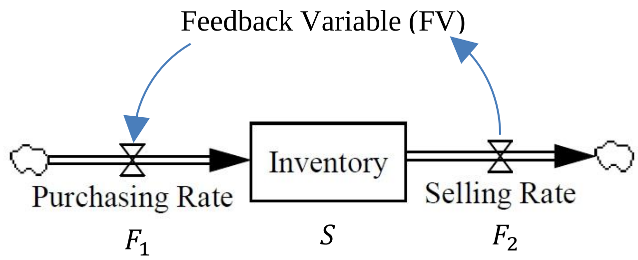


Figure 4.1 A Sample Dynamic Flow System.

The Purchasing Rate at time t in the system can be denoted as:

$$F_1^t = f(FV^t), \tag{4.4}$$

where $f(FV^t)$ is a function of FV^t, the formulation can be modified according to practical situations. The FV at time t is a function of Selling Rate at time t:

$$FV^t = f(F_2^t). \tag{4.5}$$

The Inventory Level (S^t) at time t is affected by the inventory inflow and outflow, and it can be denoted as:

$$S^t = \int_{t_0}^{t}(F_1^\tau - F_2^\tau)d\tau + S^{t_0}, \text{ or} \qquad \text{(Continuous Time Model)} \tag{4.6}$$

$$S^t = F_1^t - F_2^t + S^{t-1} = \sum_{\tau=1}^{t}(F_1^\tau - F_2^\tau) + S^{t_0}, \qquad \text{(Discrete Time Model)} \tag{4.7}$$

where t_0 is the simulation start time.

This example presents the mathematical feasibility of the SD application to a typical dynamic flow system. By generalizing this idea, a wide range of applications can be modeled by this approach.

4.1.2. Temporal and Systemic Risks (Dynamic Bayesian Networks)

Dynamic Bayesian Networks (DBNs) are responsible for the temporal and systemic risks modeling and analysis in a DFBN. The top-level SD model uses the probability values for certain variables in the DBNs to initiate further dynamic feedback flow calculations. Modified from the pioneering model by Pearl (1986), Kjaerulff & Madsen (2008) state that, for a traditional Bayesian Belief Network (BBN), the joint probability distribution with a set of variables $X = \{X_1, X_2, \dots, X_n\}$,

$$P(X) = \prod_{i=1}^{n} P(X_i | X_{pa(i)}), \tag{4.8}$$

where $pa(i) \subseteq \{1,2,\dots,i-1\}$ for all $i = 1,2,\dots,n$.

A DBN is a BBN that enables systemic risk modeling in a temporal situation (Dagum et al., 1992). For example, the DBN in Figure 4.2(a) is a DBN with two variables A and B. There are two states for each variable. The probabilities of the states in B are affected by the states in A within the same time slice. The states in A receive feedbacks from states in B from the previous time slice. The number in the squared text box on the arrow from B to A indicates that the feedback from B is sent to A in the next 1 time slice. When expanding the compact DBN for n time slices, an unrolled DBN can be obtained in Figure 4.2(b).

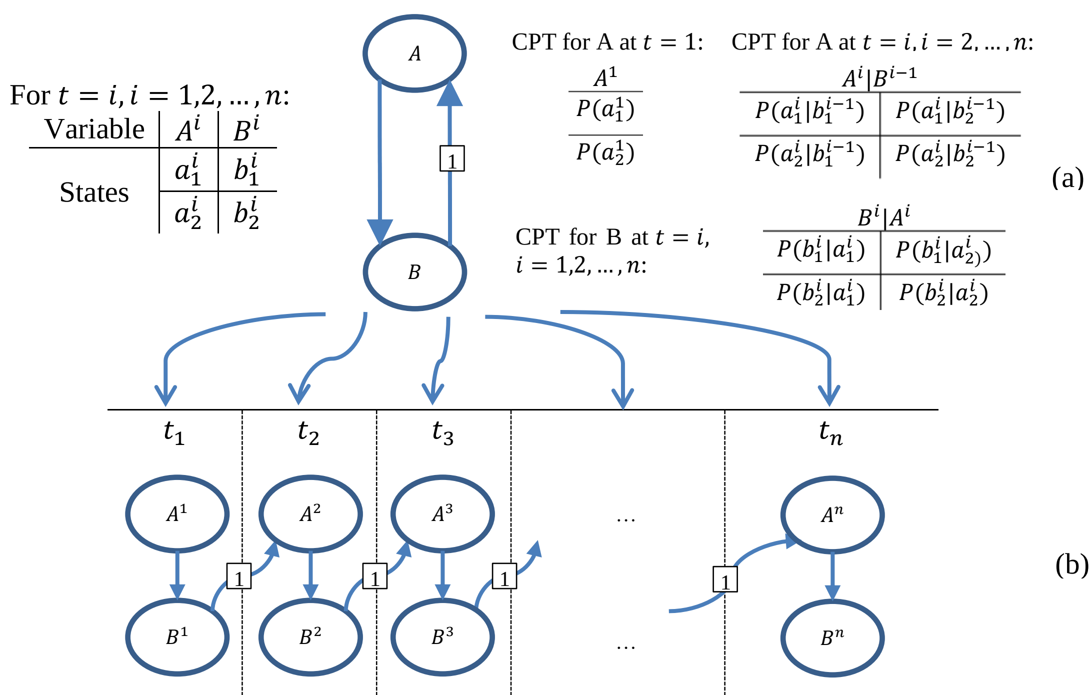


Figure 4.2 (a) A Sample DBN; and (b) the Unrolled Form of the DBN.

The joint probability distribution of the DBN in Figure 4.2 can be denoted as:

$$P(X_A, X_B) = P(A^1)P(B^1|A^1)P(A^2|B^1)P(B^2|A^2)$$
$$\dots P\left(A^i \middle| B^{i-1}\right)P\left(B^i \middle| A^i\right) \dots P(A^n|B^{n-1})P(B^n|A^n). \quad (4.9)$$

where $i = 1,2, \dots, n$, X_A, X_B are the sets of nodes, $X_A = \{A^1, A^2, \dots, A^n\}, X_B = \{B^1, B^2, \dots, B^n\}$.

The probability values for the states of the variables at time t can be calculated as:

$$P\left(a_j^t\right) = \sum_{k=1}^{2} P(a_j^t|b_k^{t-1})P(b_k^{t-1})\, for\ t = 2,3, \dots, n; j = 1,2; \quad (4.10)$$

$$P\left(b_j^t\right) = \sum_{k=1}^{2} P(b_j^t|a_k^t)P(a_k^t)\,, for\ t = 1,2, \dots, n; j = 1,2. \quad (4.11)$$

In order to establish Directed Acyclic Graphs (DAGs) and ultimately Dynamic Bayesian Networks (DBNs), causal relationships between the risk factors in the DAGs and DBNs can be determined by a manager's previous empirical works. Probability data in the Conditional Probability Tables (CPTs) in the DBNs can be generated from the database of industrial operations. However, it is usually difficult to gain data in the form of CPTs. Thus, a verbal-numerical mapping method, which is introduced by Vick (2002), is applied to acquire the conditional probability data. Verbal-numerical mapping is an alternative method that can generate numerical probability values from structured verbal descriptors. For example, Table 4.1 is one of the verbal-numerical mapping tables by Vick (2002). If the occurrence of an event is identified by a supply chain manager, and it is linguistically expressed as "Very Probable", then, from Table 4.1, the corresponding probability equivalent value can be found as 80%. By using this method, all probability values in the CPTs can be defined as linguistic expressions, and then be translated to quantitative values.

Table 4.1 A Sample Numerical Responses Table. (Source: Vick, 2002)

Linguistic Expression	Probability Equivalent (%)
Almost Certain	90
Very High Chance	90
Very Likely	85
High Chance	80
Very Probable	80
Very Possible	80
Likely	70
Probable	70
Even Chance	50
Medium Chance	50
Possible	40
Low Chance	20
Unlikely	15
Improbable	15
Very Low Chance	10
Very Unlikely	10
Very Improbable	5
Almost Impossible	2

4.1.3. Dynamic Flow Bayesian Network (DFBN)

A Dynamic Flow Bayesian Network (DFBN) is introduced to associate the temporal and systemic risk analysis with the dynamic flow system. The previous two sections provide the mathematical expressions for the foundation of the DFBN, which are SD and DBNs. The connection between SD and DBNs can be defined as Risk Affected Variables (RAVs). The term RAV is created to present the variables that are affected by the risks, i.e. the outputs from the DBNs. The impact of the risk and the probability of risk occurrence are two key elements in the presentation of RAVs. The impact of the risk is concluded from the operations of a system. The probability of risk occurrence is obtained from the selected variables in the DBNs, and we call these variables as Output Variables (OVs). The general form of an RAV can be denoted by the following equation:

$$\text{RAV} = f(impact, P(OV)), \tag{4.12}$$

where $P(OV)$ is the probability of the OV in the DBN. The formulation of an RAV can be different when considering different problem domains. A simple demonstration of Equation 4.12 is the expected value formula:

$$\mathrm{RAV}^t = X_{OV_j}^t P(OV_j^t), \tag{4.13}$$

where $X_{OV_j}^t$ is the impact of state j of OV occurring at time t, $P(OV_j^t)$ is the probability value of state j of OV at time t in the DBN. For a sample DFBN illustrated in Figure 4.3, node B is selected as the OV of the DBN. The FV is affected by the probability of the risk occurring $P(b_j^t)$, the impact of the risk (X_{OV}) and Selling Rate (F_2). We assume that:

a) time is discrete; and
b) F_2 is a positive constant C_2 at any time, respectively; and
c) variables in the SD affect each other linearly.

The FV in this system is affected by the output from a DBN, so it can be treated as an RAV. According to Equation 4.12, we can define FV^t as:

$$FV^t = f\big(f(B, X_B), f(F_2)\big) = f\big(E[X_B^t], f(F_2)\big)$$

$$= E\big[X_{B_\tau}\big] + f(F_2) = X_{B_j}^t P(b_j^t) + C_2. \tag{4.14}$$

Assume that the Purchasing Rate (F_1) is C_1 times of FV, and C_1 is a constant. Then, the Purchasing Rate at time t is:

$$F_1^t = f(FV^t) = C_1 FV^t = C_1 f\big(f(B, X_B), f(F_2)\big) = C_1 f\big(E[X_B^t], f(F_2)\big)$$

$$= C_1\left(E\big[X_{B_\tau}\big] + f(F_2)\right) = C_1 X_{B_j}^t P(b_j^t) + C_1 C_2. \tag{4.15}$$

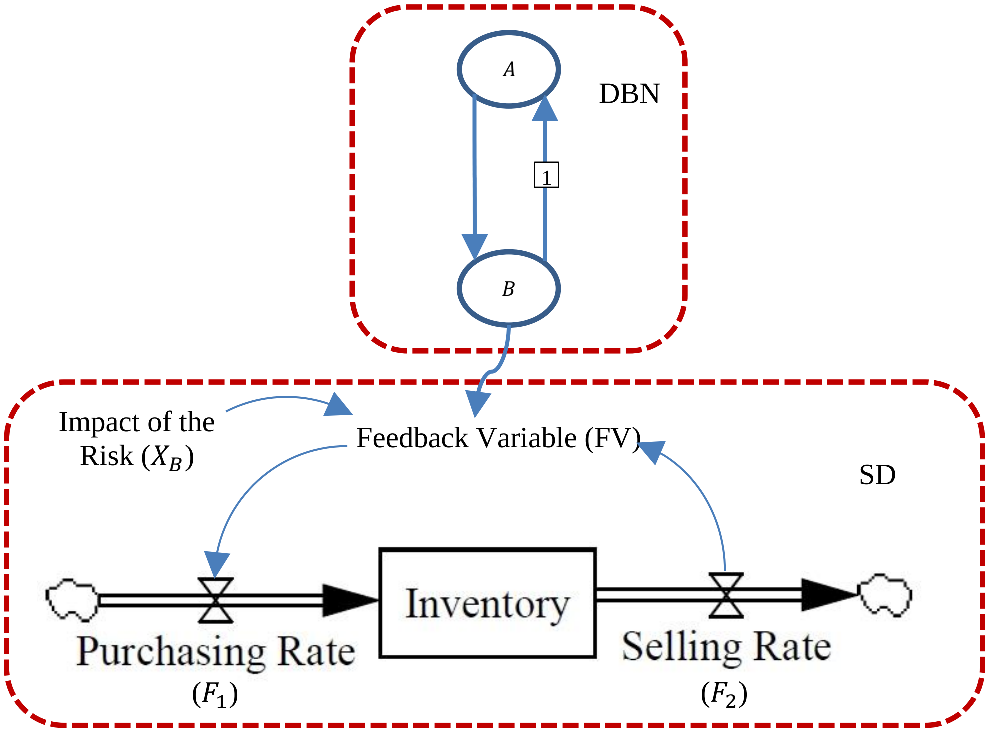


Figure 4.3 A Sample DFBN with One RAV.

Finally, from Equation 4.7, the Inventory Level at time t (S^t) can be denoted as the sum of the difference between its inflow rate (F_1) and outflow rate (F_2) at each time slice τ:

$$S^t = \sum_{\tau=1}^{t}(F_1^\tau - F_2^\tau) + S^{t_0} = \sum_{\tau=1}^{t}\left(C_1 X_{B_j}^\tau P(b_j^\tau) + C_1 C_2 - C_2\right) + S^{t_0}. \tag{4.16}$$

In summary, in the DFBN with a single RAV, as shown in Figure 4.3, the flow rates (FV^t and F_1^t) and inventory level (S^t) at any simulation time t can be obtained by using Equations 4.14 - 4.16.

4.2. Development of Temporal Uncertainty Analysis Techniques

A system in the real business world may encounter more complicated situations, such as evidence of risks at a specific time, mitigations against risks, inventory replenishment plans and system disruption risk analysis. This section provides a view of how a DFBN can be modified to adapt to different system scenarios.

4.2.1. Evidence

In a DFBN, some observation of events can be found inside the DBNs. These observations are called evidence of the network. Based on the d-separation theory that describes the information propagation in BBNs by Pearl (1988), several scholars develop the idea of evidence (Nielsen & Jensen, 2009; Kjaerulff & Madsen, 2008; Fung & Chang, 2013). For instance, in the DBN in Figure 4.2, the second state of variable A at time $t = 2$ is observed to be true, then $P(a_1^2) = 0$ and $P(a_2^2) = 1$ should replace the original conditional probability generated by the causal relationship in the DBN. The evidence in the DBNs can be seen as the new conditions for the original probability values. As a DBN needs to export probability values for some states of the variables to the SD model (i.e. $P(b_j^t)$ in Equation 4.11), the evidence observed in the network may alter the output values. By applying the evidence models indicated by Kjaerulff & Madsen (2008) into the model in Figure 4.2, we assume that the evidence $\varepsilon_{v_z}^{\tau}$ is observed at time τ for state z of the variable v. The updated output probability value for b_j^t can be denoted as:

(1) if the observed variable $v = a$, then

$$P\left(b_j^t \middle| \varepsilon_{v_z}^{\tau}\right) = \sum_{k=1}^{2} P\left(b_j^t \middle| a_k^t\right) \sum_{l=1}^{2} P\left(a_k^t \middle| b_l^{t-1}\right) \sum_{m=1}^{2} P\left(b_l^{t-1} \middle| a_m^{t-2}\right) \dots \sum_{z=1}^{2} P\left(b_y^{\tau} \middle| \varepsilon_{v_z=e}^{\tau}\right) P\left(\varepsilon_{v_z}^{\tau}\right),$$

$$for\ t = 1,2, \dots, n; \tau \le t; k, l, m, \dots, z \in \{1,2\}; \tag{4.17}$$

(2) if the observed variable $v = b$, then

$$P\left(b_j^t \middle| \varepsilon_{v_z}^{\tau}\right) = \sum_{k=1}^{2} P\left(b_j^t \middle| a_k^t\right) \sum_{l=1}^{2} P\left(a_k^t \middle| b_l^{t-1}\right) \sum_{m=1}^{2} P\left(b_l^{t-1} \middle| a_m^{t-2}\right) \dots \sum_{z=1}^{2} P\left(a_y^{\tau+1} \middle| \varepsilon_{v_z=e}^{\tau}\right) P\left(\varepsilon_{v_z}^{\tau}\right),$$

$$for\ t = 1,2, \dots, n; \tau \le t; k, l, m, \dots, z \in \{1,2\}. \tag{4.18}$$

If the probability of evidence occurrence $P\left(\varepsilon_{v_z}^{\tau}\right) = 1$ for any state, then it is called hard evidence; if $P\left(\varepsilon_{v_z}^{\tau}\right) \in [0,1)$, then it is said the soft or virtual evidence is provided.

In another case, if the conditional probability $P(\varepsilon_{v_z}^{\tau}|b_j^t)$ is known, then the output probability value under evidence $\varepsilon_{v_z}^{\tau}$ can be directly obtained by using Bayes' Rule:

$$P\left(b_j^t \middle| \varepsilon_{v_z}^{\tau}\right) = \frac{P(\varepsilon_{v_z}^{\tau}|b_j^t)P(b_j^t)}{P(\varepsilon_{v_z}^{\tau})}, \tag{4.19}$$

because $P(b_j^t)$ and $P(\varepsilon_{v_z}^{\tau})$ are already known. As a dynamic model, a DFBN can insert multiple evidence in a single DBN in different time slices and variables. For example, if N evidences $\mathcal{E}^N = \{ {}^1\varepsilon_{v_z}^{\tau}, {}^2\varepsilon_{v_z}^{\tau}, \dots, {}^N\varepsilon_{v_z}^{\tau}\}$ can be provided for the DFBN in Figure 4.3, and if those evidences are independent, then, the probability of state b_j at time t can be denoted as:

$$P\left(b_j^t \middle| \mathcal{E}^N\right) = \frac{P(\mathcal{E}^N|b_j^t)P(b_j^t)}{P(\mathcal{E}^N)} = \frac{P(b_j^t)\prod_{n=1}^{N} P({}^n\varepsilon_{v_z}^{\tau}|b_j^t)}{\prod_{n=1}^{N} P({}^n\varepsilon_{v_z}^{\tau})}. \tag{4.20}$$

In the application of the DFBN, evidence data are mainly collected from business operators' empirical knowledge about the fluctuating environment. The verbal-numerical mapping method by Vick (2002) can be utilized to transfer linguistic understanding of events to probability values of the evidence $\varepsilon_{v_z}^{\tau}$.

4.2.2. Mitigations

When risks are identified in some processes, mitigations can be deployed to minimize the impact or disruptions to the functionality of the system. Chopra & Meindl (2010) provide a list of mitigation plans that neutralize the influence of supply chain risks (Table 3.2). When a mitigation plan is deployed to a process, the probability of the occurrence of the risk should normally be reduced. A mitigation plan can also be seen as an observation of event inside a system, i.e., evidence that is introduced in Section 4.2.1. Similar to Equations 4.17 – 4.20, the following mitigation equations are based on the evidence models indicated by Kjaerulff & Madsen (2008). Thus, the probability of the output variable can be denoted as the original value under the condition of the mitigation. For the DFBN illustrated in Figure 4.3, if there is a mitigation plan for state z of the variable v at time τ, the probability for state j of the output variable B at time t can be presented as:

(1) if the mitigation is for variable $v = a$, then

$$P\left(b_j^t \middle| \mu_{v_z}^\tau\right) = \sum_{k=1}^2 P\left(b_j^t \middle| a_k^t\right) \sum_{l=1}^2 P\left(a_k^t \middle| b_l^{t-1}\right) \sum_{m=1}^2 P\left(b_l^{t-1} \middle| a_m^{t-2}\right) \dots \sum_{z=1}^2 P\left(b_y^\tau \middle| \mu_{v_z=i}^\tau\right) P\left(\mu_{v_z}^\tau\right),$$

$$for\ t = 1,2,\dots,n; \tau \le t; k,l,m,\dots,z \in \{1,2\}; \tag{4.21}$$

(2) if the mitigation is for variable $v = b$, then

$$P\left(b_j^t \middle| \mu_{v_z}^\tau\right) = \sum_{k=1}^2 P\left(b_j^t \middle| a_k^t\right) \sum_{l=1}^2 P\left(a_k^t \middle| b_l^{t-1}\right) \sum_{m=1}^2 P\left(b_l^{t-1} \middle| a_m^{t-2}\right) \dots \sum_{z=1}^2 P\left(a_y^{\tau+1} \middle| \mu_{v_z}^\tau\right) P\left(\mu_{v_z}^\tau\right),$$

$$for\ t = 1,2,\dots,n; \tau \le t; k,l,m,\dots,z \in \{1,2\}. \tag{4.22}$$

If the probability value $P(\mu_{v_z}^\tau | b_j^t)$ can be collected, then $P\left(b_j^t \middle| \mu_{v_z}^\tau\right)$ can be calculated by using Bayes' Rule:

$$P\left(b_j^t \middle| \mu_{v_z}^\tau\right) = \frac{P(\mu_{v_z}^\tau | b_j^t) P(b_j^t)}{P(\mu_{v_z}^\tau)}. \tag{4.23}$$

If N mitigation plans $\mathrm{M}^N = \left\{ {}^1\mu_{v_z}^\tau, {}^2\mu_{v_z}^\tau, \dots, {}^N\mu_{v_z}^\tau \right\}$ are deployed, then, the probability of state b_j at time t can be denoted as:

$$P\left(b_j^t \middle| \mathrm{M}^N\right) = \frac{P(\mathrm{M}^N | b_j^t) P(b_j^t)}{P(\mathrm{M}^N)} = \frac{P(b_j^t) \prod_{n=1}^N P({}^n\mu_{v_z}^\tau | b_j^t)}{\prod_{n=1}^N P({}^n\mu_{v_z}^\tau)}. \tag{4.24}$$

For the probability values of mitigation plans, the verbal-numerical mapping method by Vick (2002) can also be applied if insufficient industry data are provided.

4.2.3. Value-at-Risk

The link between DBNs and SD in a DFBN model can be called as the risk quantification method. RAV is the key variable to show the risk quantifying process. Equation 4.12 describes the establishment of an RAV. It is a function of the probability of the OV and the impact of the OV. The expected value method in Equation 4.13 is one of the most straightforward methods to obtain the value of an RAV. However, in complex systems, such as supply chains, rare but severe events, i.e. disruptions, may cause particular effects than more frequent and mild events. Supply chain disruptions, such as strikes, hurricanes, terrorist attacks and fires are identified as Value-at-Risk (VaR) type risks. Extreme value theory (EVT) is a method to describe and analyze VaR type risks (Yang, 2007). EVT has been widely applied in the financial industry as a risk assessment tool (Gilli & Kellezi, 2006; Marimoutou et al., 2009; Gencay & Selcuk, 2004; McNeil & Frey, 2000). Other popular research domains, such as genetics (Joyce et al., 2008), sonar reverberation (La Cour, 2004) and environmental monitoring (Towler et al., 2010) also use EVT as their statistical model for quantitative risk analysis.

The impact of disruptive events is assumed to be the financial loss caused by those events in EVT (Gilli & Kellezi, 2006; Yang, 2007). Generalized extreme value distributions (GEVD) are a series of generalized distributions that can be used in EVT (Castillo et al., 2005). The probability density function (PDF) and cumulative distribution function (CDF) for the maximum GEVD are denoted as:

$$f(x;\lambda,\delta,\xi) = \frac{1}{\delta}\exp\left(-\left[1-\xi\left(\frac{x-\lambda}{\delta}\right)\right]^{\frac{1}{\xi}}\right)\left[1-\xi\left(\frac{x-\lambda}{\delta}\right)\right]^{\frac{1}{\xi}-1}, \tag{4.25}$$

$$F(x;\lambda,\delta,\xi) = \exp\left(-\left[1-\xi\left(\frac{x-\lambda}{\delta}\right)\right]^{\frac{1}{\xi}}\right). \tag{4.26}$$

When $\xi \to 0$, the PDF and CDF can be denoted as:

$$f(x;\lambda,\delta,0) = \frac{1}{\delta}\exp\left[-\exp\left(\frac{x-\lambda}{\delta}\right)\right]\exp\left(\frac{x-\lambda}{\delta}\right), \tag{4.27}$$

$$F(x;\lambda,\delta,0) = \exp[-\exp\left(\frac{x-\lambda}{\delta}\right)]. \tag{4.28}$$

For the Equations 4.25 - 4.28, x is the impact of disruptions, ξ is the shape parameter, δ is the scale parameter and λ is the location parameter. Maximum likelihood estimation (MLE) and the method of moments can be utilized to estimate the parameters of GEVD (Ravindran & Warsing, 2012). Another GEVD parameter estimation method is suggested by Hosking et al. (1985).

Acting as the linkage between DBNs and SD, the probability values of VaR, which is the left-hand-side of Equation 4.26 and 4.28, are obtained from the upstream DBNs (Equation 4.10 and 4.11). Then, the variable x in Equation 4.26 and 4.28 can be generated through the functions. DBNs can generate a series of probability values regarding different time slices. Thus, the output of VaR is x_i's, where i is the notation of time. Finally, x_i's send the risk-considered information as input flows for the following SD simulation process.

4.2.4. Mathematical Optimization

Mathematical optimization is the technique of acquiring the best solution to mathematical problems (Snyman, 2005). Normally, there are two processes of the mathematical optimization: (a) optimization problem formulation and (b) solution of the constrained optimization problem. The general mathematical form of a mathematical optimization problem can be denoted as:

$$minimize\ f(\mathbf{x}), \mathbf{x} = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n, \tag{4.29}$$

subject to the constraints:

$$g_j(\mathbf{x}) \leq 0,\ \ j = 1,2, \dots, m, \tag{4.30}$$

$$h_j(\mathbf{x}) = 0,\ \ j = 1,2, \dots, r, \tag{4.31}$$

where $\mathbb{R}^n$ is the n-dimensional Euclidean space, x_i in the column vector $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ are the decision variables, $f(\mathbf{x})$ denotes the objective function, $g_j(\mathbf{x})$ is called the inequality constraint functions and $h_j(\mathbf{x})$ represents equality constraint functions. The optimum vector $\mathbf{x}$ that solves the mathematical optimization problem in Equations 4.29 – 4.31 can be denoted as $\mathbf{x}^*$. The corresponded optimal objective value is $f(\mathbf{x}^*)$. For the maximization problems, the objective function (Equation 4.29) can be reformulated by minimizing the original function multiplying by -1.

In real applications, mathematical optimization problems may involve multiple objectives to be satisfied simultaneously. For instance, an investment portfolio manager needs to consider several objectives, such as risk, payback period, and opportunity cost other than sole profitability for a financial product. The objectives for a particular case usually conflict each other (Rangaiah, 2009). Compromise on some objectives may be required when the optimum for one objective is achieved. The Pareto optimal solution is one of the approaches that can solve a multi-objective optimization problem. For a multi-objective optimization problem, the objective function can be formulated as:

$$minimize\ \left(f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})\right), k \geq 2. \tag{4.32}$$

The vector $\mathbf{x}^{\boldsymbol{P}}$ is said to be a Pareto-optimal solution for the multi-objective optimization problem if and only if, for all $i \in k$, there is no $\mathbf{x}$ such that $f_i(\mathbf{x}) \leq f_i\left(\mathbf{x}^{\boldsymbol{P}}\right)$. The optimal value is $f(\mathbf{x}^{\boldsymbol{P}})$. It is hard to satisfy all optimal values for each objective in a multi-objective optimization problem. Thus, regarding with k objectives, the problem can be transformed into a series of single objective problems with k different solutions. The set of solutions obtained by the single objective problems is called a Pareto front, if $k = 2$, or a Pareto surface, if $k > 2$ (Kim & De Weck, 2005).

Another approach to solving the multi-objective optimization problem is the weighted sum method. The weighted sum method is continuously used to provide multiple solutions by varying the weights for different objectives with distinct preferences (Marler & Arora, 2010). Using the weighted sum method to solve the multi-objective optimization problem in Equation 4.32 requires assigning scalar weight w_i for each of the original objective functions $f_i(\mathbf{x})$. Then, by summing the weighted objective functions up, the new objective function is:

$$minimize \ \sum_{i=1}^{k} w_i f_i(\mathbf{x}), k \geq 2. \tag{4.33}$$

In addition, if all of w_i's are positive, Equation 4.33 is always Pareto optimal (Zadeh, 1963).

The weighted sum method has been applied in solving the multi-objective optimization problem in various domains. Zhang et al. (2009) and Christensen et al. (2008) use this approach studying the Multiple Input Multiple Output (MIMO) broadcasting problems. Murata et al. (1996) apply the weighted sum method in the flowshop scheduling optimization problem. Other areas also show vast applications of the weighted sum method, such as load frequency control in electrical engineering (Naidu et al., 2014), carbon dioxide capture in chemical engineering (Kangwanpongpan et al., 2012), power consumption optimization in energy system engineering (Khan, 2009) and cellular system allocation optimization in telecommunication engineering (Wang & Vandendorpe, 2011). In our DFBN model, the weighted sum method is deployed to solve the multi-objective problem. The following section demonstrates a preliminary application of the DFBN for a three-stage supply chain.

4.3. An Application of Modeling System on Supply Chain Analysis

A supply chain connects a series of production and distribution of merchandise processes. With the pursuit of optimizing allocation of production materials and target markets, supply chain components are usually separated. A typical five-stage supply chain consists of raw material suppliers, manufacturers, wholesalers, retailers and customers. Dynamic flows of capital, commodity, information and people are formed to realize the profit of business entities inside a supply chain. This section introduces an initial application scenario for the proposed DFBN method.

4.3.1. Background and Problem Identification

To enhance the performance of a local delicatessen, a manager investigates the supply chain of a popular food of their store. A three-stage supply chain is presented to study the effect of dynamic risks on inventory levels of the delicatessen (the retailer). From the market investigation, the manager found that the profitability, the risk of stock out of the food, product freshness and holding cost can be balanced if the inventory level of the store can be held to 150 for the next 12 months. The manager builds the causal relationships of systemic probabilistic risks in a DBN model and forward flows and feedbacks in an SD structure (Figure 4.4). The SD proportion of the illustration is built with Vensim PLE (http://vensim.com/), and the DBN part is constructed with GeNIe (http://www.bayesfusion.com/). Two DBNs are involved in this case study: the left-hand-side wholesaler related risks and the right-hand-side retailer related risks. The risk-related variables in the DBNs refer to the risk metrics from the SCRIPT (Figure 3.3). Both DBNs propagate the supply chain risk information into the feedback flows in the SD. Table 4.2 summarizes the different types of variables in Figure 4.4.

Table 4.2 Variables and Their Attributes for the Sample Supply Chain.

Model of Variable	Variable Type	Variable Name	Variable Attribute
System Dynamics (SD)	Stock	Wholesaler Inventory (WI)	$WI^i = WI^0 + \sum_{j=1}^{i}(WPR^j - RPR^j)$, $WI^0 = 2000\ Units$
		Retailer Inventory (RI)	$RI^i = RI^0 + \sum_{j=1}^{i}(RPR^j - CPR^j)$, $RI^0 = 200\ Units$
	Rate	Wholesaler Procurement Rate (WPR)	$WPR^i = \begin{cases} WSR^i + 100, & RI^i \leq 1950 \\ WSR^i, & Otherwise \end{cases}$
		Retailer Procurement Rate (RPR)	$RPR^i = \begin{cases} RSR^i + 50, & RI^i \leq 150 \\ RSR^i, & Otherwise \end{cases}$
		Customer Purchasing Rate (CPR)	$CPR^i = P(COL^i_{High}) * 100 + (1 - P(COL^i_{High})) * 50$

Table 4.2 Variables and Their Attributes for the Sample Supply Chain (Continued).

Model of Variable	Variable Type	Variable Name	Variable Attribute
System Dynamics (SD)	Feedback	Wholesaler Sales Rate (WSR)	$WSR^i = RPR^i - REUOR^i$
		Retailer Sales Rate (RSR)	$RSR^i = CPR^i - CEUOR^i$
		Retailer's Expected Unfilled Order Rate (REUOR)	$REUOR^i = P(RUO^i_{Yes}) * RUOA^i$
		Retailer's Unfilled Order Amount (RUOA)	Empirical Data
		Probability of Retailer's Unfilled Order (RUO)	States: High/Low
		Customer's Expected Unfilled Order Rate (CEUOR)	$CEUOR^i = P(CUO^i_{Yes}) * CUOA^i$
		Customer's Unfilled Order Amount (CUOA)	Empirical Data
		Probability of Customer's Unfilled Order (CUO)	States: High/Low
		Probability of Customer Order Level (COL)	States: High/Low
Dynamic Bayesian Network (DBN)	Wholesaler Related Risks	Manufacturer Supplying Difficulties (MSD)	States: High/Low
		Retailer Capital Amount (RCA)	States: High/Low
		Wholesaler Expected Procurement (WEP)	States: High/Low
		Retailer Order Level (ROL)	States: High/Low
		Wholesaler Inventory Level (WIL)	States: High/Low
		Retailer Unfilled Orders (RUO)	States: Yes/No
		Wholesaler Long Term Competitiveness (WLTC)	States: Strong/Weak
	Retailer Related Risks	Wholesaler Supplying Difficulties (WSD)	States: High/Low
		Customer Capital Amount (CCA)	States: High/Low
		Retailer Expected Procurement (REP)	States: High/Low
		Customer Order Level (COL)	States: High/Low
		Retailer Inventory Level (RIL)	States: High/Low
		Customer Unfilled Orders (CUO)	States: Yes/No
		Retailer Long Term Competitiveness (RLTC)	States: Strong/Weak

4.3.2. Dynamic Flow Bayesian Network (DFBN) Modeling

A DFBN utilizes the temporal probability risk information from the DBNs, and propagates the information into SD. The arrows between variables from SD and DBN connect the input values of the SD variables and the result obtained from the DBNs. The arrows with a plus or minus sign in the SD part indicate that variables have positive or negative feedbacks (Georgiadis et al., 2005).

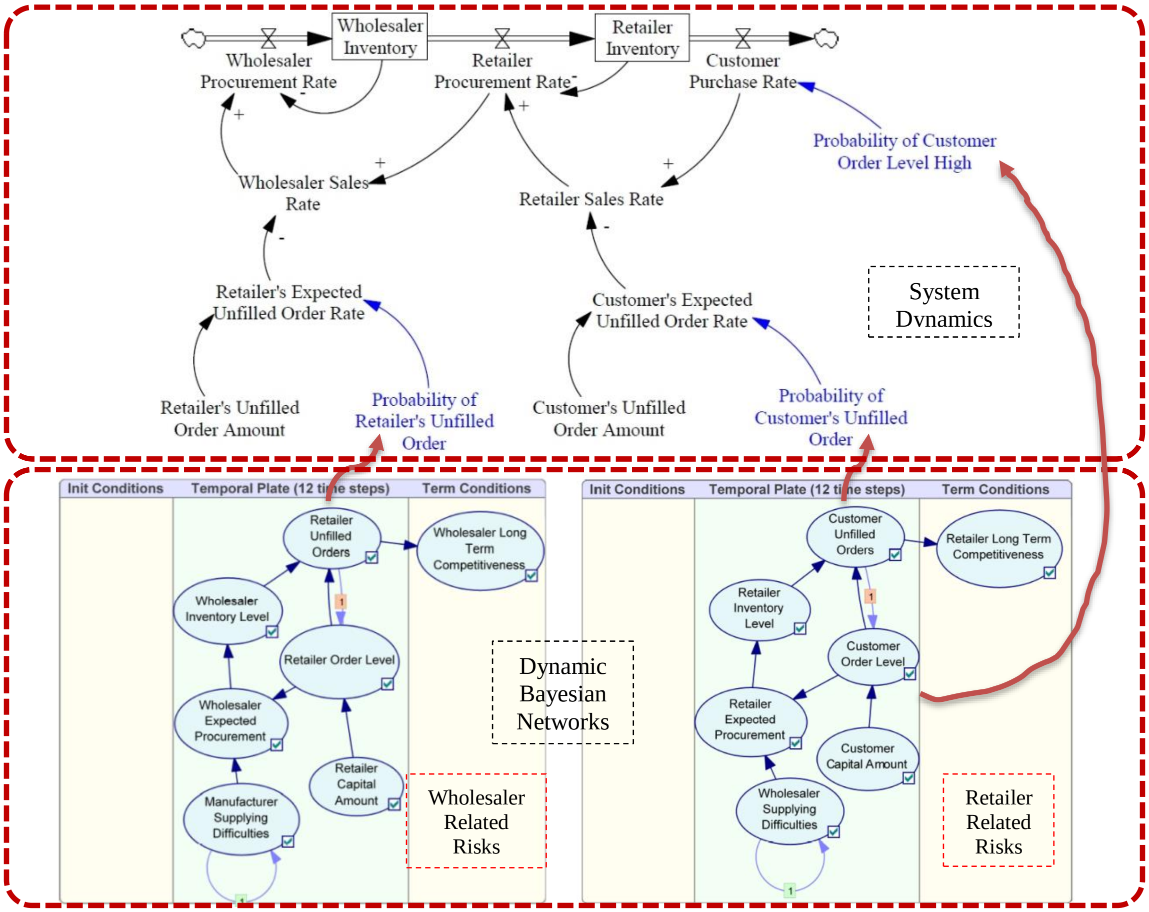


Figure 4.4 DFBN Diagram for a Local Delicatessen Supply Chain.

4.3.2.1. Dynamic Bayesian Networks (DBNs)

In Figure 4.4, nodes in the DBNs' initial (Init) conditions only affect their descendants at the first time slice. For the nodes in the termination (Term) conditions, their probabilities only depend on the last time slice of their parent nodes. In the temporal plate, nodes interact with their parent and decedent nodes in every time slice. If the arrow between the parent nodes and their descendant nodes in the temporal plate has no numbers on it, the nodes interact with each other from the same time slice. For the arrows with a number n on it, the parent node affects the descendant node at the next n time slice. In order to determine the inventory strategy for the store in the next 12 months, the simulation is setup for $T = 12$ time slices with each of them representing one month. The states of the variables in DBNs are presented in Table 4.2, and the probability of state occurrence are stored in the CPTs (Figure 4.5 and Figure 4.6). The values in CPTs in the DBNs are obtained through verbal-numerical mappings due to the limitation of accessing the historical conditional probability values. Verbal-numerical mapping is a method that can explain likelihood judgment by using verbal descriptors. Then, numerical probability values can be translated from the determined verbal descriptors (Vick, 2002). For example, if the manager decides $P(CCA_1^2)$ is "relatively high", the probability value "0.8" corresponds to this verbal descriptor can be found in the verbal-numerical mappings tables. The verbal descriptors are from manager's experience and the corresponded probability values are listed in the CPTs in Figure 4.5 and 4.6.

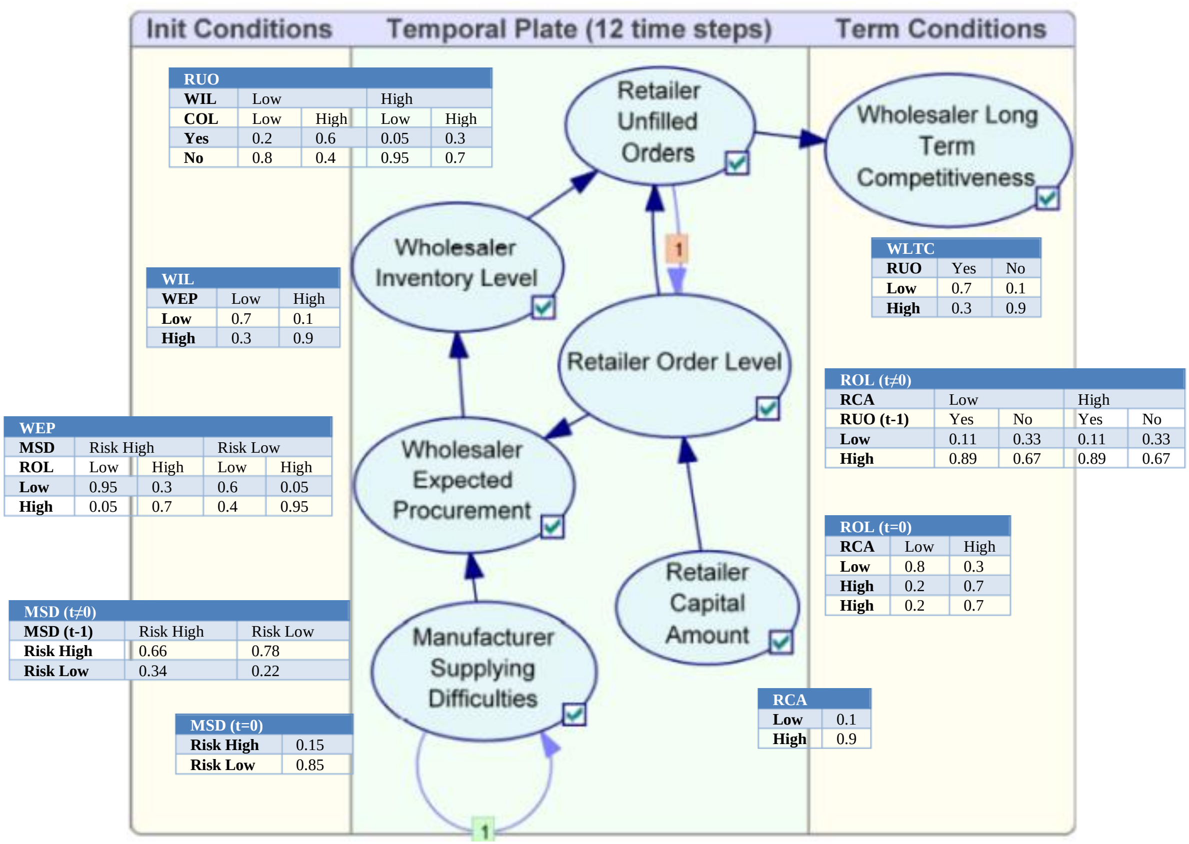


Figure 4.5 DBN for Retailer Related Risks with CPTs.

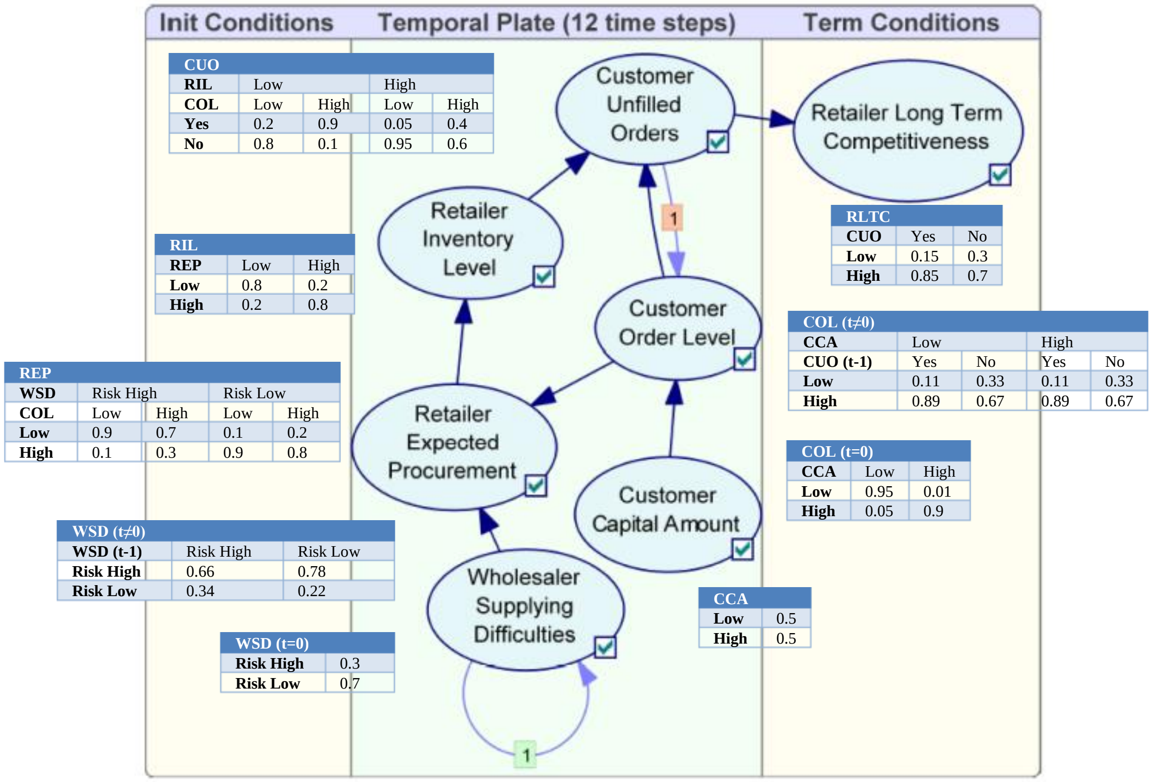


Figure 4.6 DBN for Wholesaler Related Risks with CPTs.

The output variable for retailer related risks is CUO^i and COL^i; for the wholesaler related risks, the output variable is RUO^i, where i represents the time slice. For instance, CUO^i is the descendant of WIL^i and ROL^i, the joint probability distribution can be denoted as:

$$P(CUO_j^i) = \sum_{k=1}^{2} P(CUO_j^i | RIL_k^i, COL_k^i) P(RIL_k^i, COL_k^i)$$

$$= \sum_{k=1}^{2} P(CUO_j^i | WIL_k^i, COL_k^i) P(RIL_k^i) P(COL_k^i)$$

$$= \sum_{k=1}^{2} P(CUO_j^i | WIL_k^i, COL_k^i) [\sum_{l=1}^{2} P(RIL_k^i | REP_l^i)\, P(REP_l^i)] \cdot$$

$$[\sum_{m=1}^{2} P(COL_k^i | CCA_m^i, CUO_m^i) P(CCA_m^i, CUO_m^i)]$$

$$= \sum_{k=1}^{2} P(CUO_j^i | WIL_k^i, COL_k^i) \cdot$$

$$\{\sum_{l=1}^{2} P(RIL_k^i | REP_l^i)\, [\sum_{n=1}^{2} P(REP_l^i | COL_n^i, WSD_n^i) P(COL_n^i, WSD_n^i)]\} \cdot$$

$$\sum_{m=1}^{2} P(COL_k^i | CCA_m^i, CUO_m^{i-1}) P(CCA_m^i) P(CUO_m^{i-1})$$

$$= \sum_{k=1}^{2} P\left(CUO_j^i \middle| WIL_k^i, COL_k^i\right) \cdot$$

$$\left\{\sum_{l=1}^{2} P\left(RIL_k^i \middle| REP_l^i\right) \left[\sum_{n=1}^{2} P\left(REP_l^i \middle| COL_n^i, WSD_n^i\right) P\left(WSD_n^i\right) P\left(COL_n^i\right)\right]\right\} \cdot$$

$$\sum_{m=1}^{2} P\left(COL_k^i \middle| CCA_m^i, CUO_m^{i-1}\right) P\left(CCA_m^i\right) P(CUO_m^{i-1})$$

$$= \sum_{k=1}^{2} P\left(CUO_j^i \middle| WIL_k^i, COL_k^i\right) \cdot$$

$$\left\{\sum_{l=1}^{2} P\left(RIL_k^i \middle| REP_l^i\right) \sum_{n=1}^{2} P\left(REP_l^i \middle| COL_n^i, WSD_n^i\right) \left[\sum_{p=1}^{2} P\left(WSD_n^i | WSD_p^{i-1}\right) P(WSD_p^{i-1})\right] \cdot\right.$$

$$\left.\left[\sum_{q=1}^{2} P\left(COL_n^i \middle| CCA_q^i, CUO_q^{i-1}\right) P\left(CCA_q^i\right) P(CUO_q^{i-1})\right]\right\} \cdot$$

$$\sum_{m=1}^{2} P\left(COL_k^i \middle| CCA_m^i, CUO_m^{i-1}\right) P\left(CCA_m^i\right) P(CUO_m^{i-1}),\ for\ i = 2, \dots, T; j = 1,2. \tag{4.34}$$

In Equation 4.34, we can notice that $P\left(CUO_j^i\right)$ at time i is calculated by using the variables from the previous time slice, i.e. WSD_n^{i-1}, CUO_q^{i-1} and CUO_m^{i-1}. It indicates that the risk information propagates through the time slices from these variables. This is the essential idea of temporal risk modeling by applying DBNs.

4.3.2.2. System Dynamics (SD)

According to Sterman (2000), FVs in an SD can be denoted by other FVs. RAVs in the SD utilizes the probability values from OVs for each time slice. Thus, for the retailer feedback flows, the customer expected unfilled order rate (CEUOR) is the multiplication of the customer unfilled order amount (CUOA) and the probability of customer unfilled orders $P(CUO)$ at each time slice i. For the wholesaler feedback flows, the retailer expected unfilled order rate (REUOR) equals the retailer unfilled order amount (RUOA) times the probability of retailer unfilled orders $P(RUO)$ at each time slice i. For the sales rates (WSR^i and RSR^i), they are the difference between the purchasing rates and the expected unfilled order rates:

$$WSR^i = RPR^i - REUOR^i \tag{4.35}$$

$$RSR^i = CPR^i - CEUOR^i \tag{4.36}$$

We assume that the delicatessen and the wholesaler use the following replenishment plan:

$$WPR^i = \begin{cases} WSR^i + 100, & RI^i \leq 1950 \\ WSR^i, & Otherwise \end{cases} \tag{4.37}$$

$$RPR^i = \begin{cases} RSR^i + 50, & RI^i \leq 150 \\ RSR^i, & Otherwise \end{cases} \tag{4.38}$$

By adapting from Equation 4.7, the inventory levels for the retailer and wholesaler are the difference between their procurement rate and the purchasing rate of their downstream entities in the supply chain:

$$WI^i = WI^0 + \sum_{j=1}^{i}(WPR^j - RPR^j) \tag{4.39}$$

$$RI^i = RI^0 + \sum_{j=1}^{i}(RPR^j - CPR^j) \tag{4.40}$$

Equation 4.35 – 4.40 form the SD model for this case study as shown in Figure 4.4. After identifying the temporal causal relationships between risk factors in DBNs and the dynamics flows in SD, the following section introduces the method that integrates DBNs and SD.

4.3.2.3. The Integrated Model

The basic idea of combining DBNs and SD to a DFBN in this case study is obtaining RAVs by using expected values. The early stage of expected value application in decision-making is presented by Simon (1959) and Edwards (1954). More recent researches still show an abundant application of the expected value theory in biology (Puigbò et al., 2008), inventory models (Xu & Zhao, 2008), reinsurance strategy (Zheng et al., 2015) and child psychiatry (White et al., 2016).

Equation 4.13 denotes the calculation of an RAV, which is the product of $P(OV_j^t)$, the probability value of state j of OV at time t in the DBN, and $X_{OV_j}^t$, the impact of state j of OV occurring at time t. There are three RAVs in this DFBN, they are CPR, REUOR and CEUOR.

For CPR^i, the OV is COL^i, which is obtained from retailer related DBN. If COL^i is in the high state (i.e. COL^i_{High}), the customers are willing to buy 100 units of food at ith month; if COL^i is in the low state (i.e. COL^i_{Low} or $1 - COL^i_{High}$), only 50 units per month are consumed. CPR^i is the expected value of customer desired purchasing rate in each state times the probability of that state happening at a specific time slice i:

$$CPR^i = P(COL^i_{High}) * 100 + \left(1 - P(COL^i_{High})\right) * 50. \tag{4.41}$$

$REUOR^i$ and $CEUOR^i$ are the expected value of their OVs and impact of the risk (Equation 4.13). For $REUOR^i$ and $CEUOR^i$, the OVs are RUO^i and CUO^i respectively. $REUOR^i$ and $CEUOR^i$ are the product of the probability of the OVs at Yes state ($P\left(RUO^i_{Yes}\right)$ and $P\left(CUO^i_{Yes}\right)$) and the impact of the Yes state of OVs occurring at time i ($RUOA^i$ and $CUOA^i$):

$$REUOR^i = P\left(RUO^i_{Yes}\right) \ * RUOA^i \tag{4.42}$$

$$CEUOR^i = \ P\left(CUO^i_{Yes}\right) * CUOA^i \tag{4.43}$$

By using Equation 4.41 – 4.43, the dynamic linkages between DBNs and SD are connected. The FVs in SD are under the influence of temporal risks generated in DBNs. For instance, Equation 4.34 is the expanded form of $P\left(CUO^i_j\right)$. Thus, by substituting $P\left(CUO^i_{Yes}\right)$ in Equation 4.43 with Equation 4.34, it becomes:

$$\begin{aligned} CEUOR^i &= CUOA^i \textstyle\sum_{k=1}^{2} P\left(CUO^i_j \middle| WIL^i_k, COL^i_k\right) \cdot \\ &\left\{\textstyle\sum_{l=1}^{2} P\left(RIL^i_k \middle| REP^i_l\right) \textstyle\sum_{n=1}^{2} P\left(REP^i_l \middle| COL^i_n, WSD^i_n\right) \left[\textstyle\sum_{p=1}^{2} P\left(WSD^i_n | WSD^{i-1}_p\right) P(WSD^{i-1}_p)\right] \cdot \right. \\ &\left. \left[\textstyle\sum_{q=1}^{2} P\left(COL^i_n \middle| CCA^i_q, CUO^{i-1}_q\right) P\left(CCA^i_q\right) P(CUO^{i-1}_q)\right]\right\} \textstyle\sum_{m=1}^{2} P\left(COL^i_k \middle| CCA^i_m, CUO^{i-1}_m\right) P(CCA^i_m) P(CUO^{i-1}_m)\,, \\ &for\ i = 2, \ldots, T. \end{aligned} \tag{4.44}$$

By using the result from Equation 4.44, the risk factors can be propagated to other dynamic flow equations (Equation 4.35 – 4.40), and the DFBN model is established.

4.3.2.4. Evidence and Mitigations

In a DFBN, some observation of probabilistic events can be found in DBNs. These observations are called evidence in the network. The evidence in the DBNs is the new condition for the original probability value. The values of OVs of DBNs can be altered if some variables are observed. Equation 4.17 – 4.20 describe the effects of evidence on the probability values of the variables. In this case study, the manager can confirm some of the events by month in either high or low state by his experience from past years (Table 4.3).

Table 4.3 Temporal Evidence in the DBNs.

Time Slices (Month)	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Evidence												
Wholesaler Supplying Difficulties	High	High				Low	Low					
Customer's Capital Amount			High			Low					High	Low
Manufacturer's Supplying Difficulties	High	High			Low				Low	Low		
Retailer's Capital Amount	High							Low	High		Low	High

In Equation 4.34, we assume that the evidence $\varepsilon^3_{CCA_{High}}$ is observed at time $i = 3$ for state $High$ of the variable CCA. The updated value for CUO^i can be denoted as:

$$
\begin{aligned}
&P\left(CUO^i_j | \varepsilon^i_{CCA_{High}}\right) = \\
&\sum_{k=1}^{2} P\left(CUO^i_j \middle| WIL^i_k, COL^i_k\right)\left\{\sum_{l=1}^{2} P\left(RIL^i_k \middle| REP^i_l\right) \sum_{n=1}^{2} P\left(REP^i_l \middle| COL^i_n, WSD^i_n\right) \cdot \right. \\
&\left.\left[\sum_{p=1}^{2} P\left(WSD^i_n | WSD^{i-1}_p\right) P\left(WSD^{i-1}_p\right)\right]\right\} \cdot \\
&\left[\sum_{q=1}^{2} P\left(COL^i_n \middle| \varepsilon^i_{CCA_{High}}, CUO^{i-1}_q\right) P\left(CCA^i_q | \varepsilon^i_{CCA_{High}}\right) P\left(\varepsilon^i_{CCA_{High}}\right) P\left(CUO^{i-1}_q\right)\right] \cdot \\
&\sum_{m=1}^{2} P\left(COL^i_k \middle| \varepsilon^i_{CCA_{High}}, CUO^{i-1}_m\right) P\left(CCA^i_m | \varepsilon^i_{CCA_{High}}\right) P\left(\varepsilon^i_{CCA_{High}}\right) P\left(CUO^{i-1}_m\right), \\
&for\ i = 3; j = Yes.
\end{aligned}
\tag{4.45}
$$

Due to $\varepsilon^3_{CCA_{High}}$ is a hard evidence, i.e. $P\left(\varepsilon^3_{CCA_{High}}\right) = 1$ and $P\left(\varepsilon^3_{CCA_{Low}}\right) = 0$, Equation 4.45 can be shortened as:

$$P\left(CUO_j^i|\varepsilon_{CCA_{High}}^i\right) = \sum_{k=1}^{2} P\left(CUO_j^i\middle|WIL_k^i, COL_k^i\right)$$

$$\left\{\sum_{l=1}^{2} P\left(RIL_k^i\middle|REP_l^i\right) \sum_{n=1}^{2} P\left(REP_l^i\middle|COL_n^i, WSD_n^i\right) \left[\sum_{p=1}^{2} P\left(WSD_n^i|WSD_p^{i-1}\right)P(WSD_p^{i-1})\right] \cdot\right.$$

$$\left.\left[\sum_{q=1}^{2} P\left(COL_n^i\middle|\varepsilon_{CCA_{High}}^i, CUO_q^{i-1}\right) P\left(CUO_q^{i-1}\right)\right]\right\}$$

$$\sum_{m=1}^{2} P\left(COL_k^i\middle|\varepsilon_{CCA_{High}}^i, CUO_m^{i-1}\right) P\left(CUO_m^{i-1}\right), for\ i = 3; j = Yes. \tag{4.46}$$

In Equation 4.46, the evidence is in the variable (CCA^3) that directly affects another variable (COL^3) in the same time slice. However, the evidence is finally transferred to a variable (CUO^3) that can deliver this information to a variable in the next time slice. It indicates that any evidence of a variable in the DBNs at any time slice influences all its descendant variables, including the ones in the following time slices.

Mitigations are the actions taken by the supply chain managers for minimizing the impact of risks. When a mitigation is deployed, the probability of the occurrence of the risk in a process is normally expected to be reduced. In the DFBN model, a mitigation is the observation of an event in the DBN. A mitigation has the similar mechanism as an evidence when applies on an event. The probability of an OV can be denoted as its original value under the condition of the mitigation (Equation 4.21). In order to mitigate the effect of WSD to the inventory stability of the delicatessen, the manager of the delicatessen is planning to find another backup supplier at the sixth month. The manager believes that this action can mitigate the risk of WSD^2 from high to low state, i.e. $\varepsilon_{WSD_{High}}^6 \rightarrow \mu_{WSD_{Low}}^6$. Thus, CUO^i at time slice 6 can be denoted as:

$$P\left(CUO_j^i|\mu_{WSD_{Low}}^i\right) = \sum_{k=1}^{2} P\left(CUO_j^i\middle|WIL_k^i, COL_k^i\right)\left\{\sum_{l=1}^{2} P\left(RIL_k^i\middle|REP_l^i\right) \sum_{n=1}^{2} P\left(REP_l^i\middle|COL_n^i, WSD_n^i\right) \cdot\right.$$

$$\left.\left[\sum_{p=1}^{2} P\left(WSD_n^i|WSD_p^{i-1}, \mu_{WSD_{Low}}^2\right)P\left(WSD_p^{i-1}\right)\right]\left[\sum_{q=1}^{2} P\left(COL_n^i|CCA_q^i, CUO_q^{i-1}\right)P\left(CCA_q^i\right) P\left(CUO_q^{i-1}\right)\right]\right\} \cdot$$

$$\sum_{m=1}^{2} P\left(COL_k^i\middle|CCA_m^i, CUO_m^{i-1}\right)P(CCA_m^i)P(CUO_m^{i-1}),\quad for\ i = 6; j = Yes. \tag{4.47}$$

As the manager can confirm that the mitigation plan ensures the risk of WSD as low, i.e. $P\left(\mu_{WSD_{Low}}^{2}\right)=1$ and $P\left(\mu_{WSD_{High}}^{2}\right)=0$, Equation 4.45 can be shortened as:

$P\left(CUO_j^i|\mu_{WSD_{Low}}^{i}\right)=$

$\sum_{k=1}^{2}P\left(CUO_j^i\middle|WIL_k^i,COL_k^i\right)\{\sum_{l=1}^{2}P\left(RIL_k^i\middle|REP_l^i\right)\sum_{n=1}^{2}P\left(REP_l^i\middle|COL_n^i,WSD_n^i\right)\left[\sum_{p=1}^{2}P(WSD_n^{i-1})\right]\cdot$

$\left[\sum_{q=1}^{2}P\left(COL_n^i\middle|CCA_q^i,CUO_q^{i-1}\right)P\left(CCA_q^i\right)P\left(CUO_q^{i-1}\right)\right]\}$

$$\sum_{m=1}^{2}P\left(COL_k^i\middle|CCA_m^i,CUO_m^{i-1}\right)P(CCA_m^i)P(CUO_m^{i-1}),\ for\ i=6;j=Yes. \qquad (4.48)$$

From Equation 4.48, it can be observed that a mitigation possesses the similar attribute with an evidence. Any mitigation of a variable in the DBNs at any time slice influences all of its descendant variables, including the ones in the following time slices.

By applying the variables and their attributes from Table 4.2, and the CPTs from Figure 4.6, a preliminary result of the OVs in the DBNs can be obtained (Table 4.4).

Table 4.4 The Time Dependent Probabilities Obtained in the DBNs of OVs.

Time Slices (Month)	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Time Dependent Probabilities												
Customer’s Unfilled Orders (Yes)	0.43	0.49	0.55	0.54	0.54	0.53	0.65	0.66	0.54	0.54	0.54	0.54
Retailer’s Unfilled Orders (Yes)	0.42	0.49	0.55	0.53	0.65	0.54	0.55	0.53	0.65	0.66	0.54	0.54
Customer’s Order Level High	0.49	0.76	0.78	0.79	0.79	0.79	0.79	0.81	0.82	0.79	0.79	0.79

4.3.2.5. Modeling Results

The manager assumes that the initial inventory level for the delicatessen is 200 units of food. Three simulations with different replenishment plans corresponding to Equation 4.38 are listed in Table 4.5 aiming to retain the required 150 units inventory level. For example, in Sim 2, if the RI^i is less than or equal to 150 units, RPR^i should be RSR^i plus an additional 50 units; if RI^i is more than 150 units, RPR^i should be exactly the amount of RSR^i.

Table 4.5 The Replenishment Plans for Each Simulation in the SD.

	Sim 1	Sim 2	Sim 3
Replenishment Plan	$RPR^i = \begin{cases} RSR^i + 50, RI^i \leq 100 \\ RSR^i, \quad Otherwise \end{cases}$	$RPR^i = \begin{cases} RSR^i + 50, RI^i \leq 150 \\ RSR^i, \quad Otherwise \end{cases}$	$RPR^i = \begin{cases} RSR^i + 50, RI^i \leq 150 \\ RSR^i + 15, Otherwise \end{cases}$

Figure 4.7 displays the delicatessen's inventory level for the next 12 months with three different replenishment plans. Assume that the time value of money is ignored and the holding cost is $10 per unit per month for the selected food, the total holding cost for RI^i from the three simulations are $15,810, $20,310 and $22,260 respectively for the next 12 months. If the objective is minimizing holding cost, the result of Sim 1 is preferable. However, from Figure 4.7, Sim 1 may not be the ideal choice for the delicatessen if maintaining the 150 units inventory level is the priority for the manager. Sim 2 and Sim 3 seem more desirable to achieve the objective of maintaining a more stable inventory. Compared with Sim 3, the holding cost for Sim 2 is relatively low, but the inventory shortage for Sim 2 is 5 months. Sim 3 only has one month of shortage. Thus, both Sim 2 and Sim 3 can be plausible solutions for this case study.

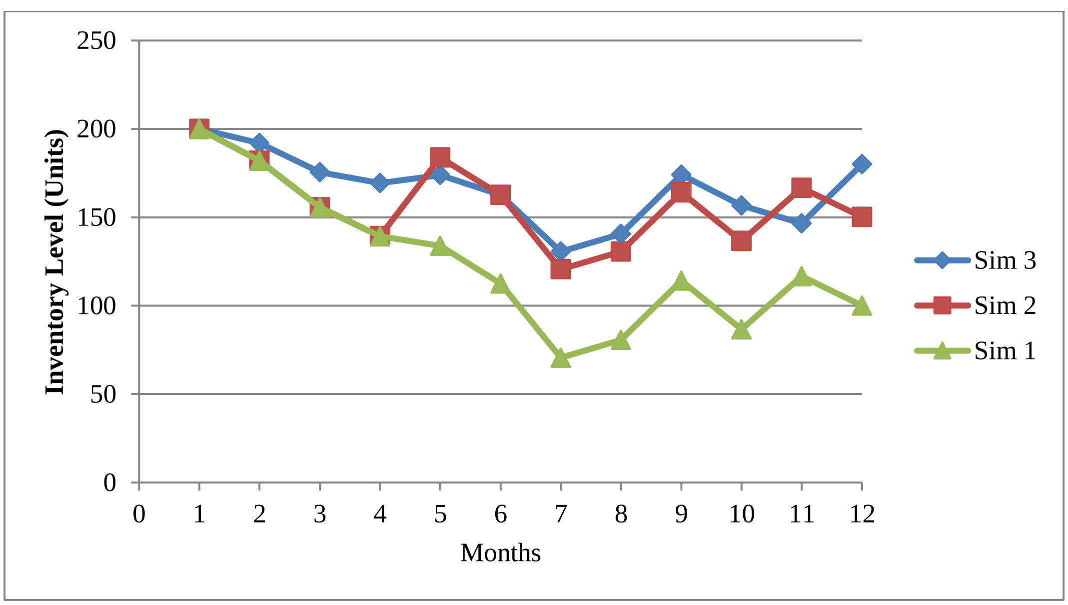


Figure 4.7 Simulation Results for the Food Market with Three Different Replenishment Plans.

A DFBN is a novel methodology that can be used to conduct risk analysis with risk enabled actual flows in a complex system. A supply chain is a complex system with flows, stocks, risks and mitigations. Thus, a supply chain system is an ideal demonstration of the application of a DFBN. The preliminary modeling result for the delicatessen case study supports the manager with developing a replenishment plan to maintain a designated inventory level. The results of the DFBN are dynamic flows of materials counted as actual units of items. This form of result provides the supply chain practitioner with a more figurative impression of the systemic influence of risks rather than the abstract probability values of event occurrence.

Although the preliminary modeling result of the DFBN is obtained in this case study, several challenges for generalizing and completing of this method remains. First, the linkage between the systemic risk model (DBN) and feedback flow simulation (SD) can be improved. A more specialized methodology can be applied to substitute the expected value method. Second, in the original DFBN, manual adjustment of replenishment parameters is required to achieve a specific supply chain objective. A more objective oriented method may increase the effectiveness and efficiency of the model. The following two sections bring two more approaches to enhance the usability of the DFBN in the business world.

4.3.3. Value-at-Risk in the DFBN

In a complex system, such as a supply chain, events that cause disruptions may be rarer but severe (Whitney, 2014). The VaR model is more suitable for the supply chain disruptions, such as strikes, hurricanes, floods, terrorist attacks, fires etc. EVT is a common method in VaR. In this model, heavy-tailed distributions, such as Gumbel, Weibull and Frechet distributions, can be used to model the impact of the disruptions (Ravindran & Warsing, 2012). According to Castillo et al. (2005), a more generalized GEVD extends the boundaries of the traditional heavy-tailed distributions. The PDF and CDF are listed in Equations 4.25 - 4.28.

The fundamental idea of applying GEVD in the DFBN can follow the procedures listed below:

i. Collect the loss data from each disruption by time.

In the delicatessen case, $REUOR_i$ and $CEUOR_i$ in the past 12 months need to be collected. The time span i should be in months. However, due to the limitation of required data provided, the manager decides to generate random data by using MATLAB normrnd() function with mean and variance from past experience for both $REUOR_i$ and $CEUOR_i$. Non-positive numbers are avoided after the random data generation.

ii. Use the disruption data estimating the parameters in the PDF of GEVD by performing MLE.

By entering the PDFs of GEVD (Equation 4.25) for $REUOR_i$ and $CEUOR_i$ respectively into the MATLAB function mle(), the parameters in the PDFs can be estimated.

iii. Apply the estimated parameters in the CDF of GEVD.

After filling in the parameters, the CDFs of $REUOR_i$ and $CEUOR_i$ can be denoted as:

$$P\left(CUO_i; \lambda_c, \delta_c, \xi_c\right) = \exp\left\{-\left[1-(-0.0370)\left(\frac{CEUOR_i-12.9552}{13.8577}\right)\right]^{\frac{1}{-0.0370}}\right\}; \tag{4.49}$$

$$P\left(RUO_i; \lambda_r, \delta_r, \xi_r\right) = \exp\left\{-\left[1-0.3361\left(\frac{REUOR_i-74.1108}{69.6784}\right)\right]^{\frac{1}{0.3361}}\right\}. \tag{4.50}$$

iv. Solve $REUOR_i$ and $CEUOR_i$ in the CDFs by entering the OV values ($P\left(CUO_i; \lambda_c, \delta_c, \xi_c\right)$ and $P\left(RUO_i; \lambda_r, \delta_r, \xi_r\right)$) generated from the DBNs (Figure 4.8).

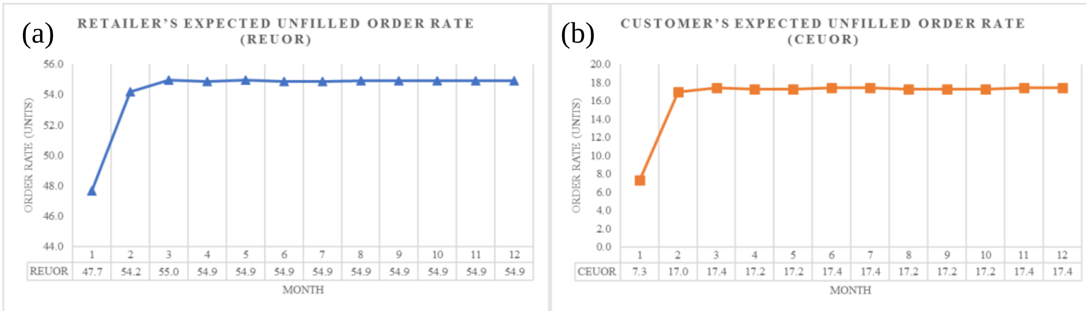


Figure 4.8 EVT Results for (a) Retailer's Expected Unfilled Order Rate (REUOR); and (b) Customer's Expected Unfilled Order Rate (CEUOR).

The advantage of applying EVT instead of the original expected value method is that EVT needs fewer assumptions than the expected value method. In the expected value method, precise prediction of the impact of the risk in the future ($X_{OV_j}^{t}$ in Equation 4.13) determines the accuracy of the RAV. However, people can hardly conduct forecasting without a mature approach. EVT is a probability-distribution-based that uses GEVD to fit past data into a distribution function. It is a reliable tool to summarize the past data for forecasting. The application of EVT remedies the inadequacy of the DBN and SD integration. In the following section, the mathematical optimization is introduced to further improve the DFBN model with a more objective-oriented method.

4.3.4. Optimization-Enabled DFBN

Mathematical optimization enables the DFBN model with a more objective-oriented method. By establishing the constraints of variables and goals of the business, the dynamic flows can be simulated within the required conditions, and finally, converged into an optimal solution under the temporal risks. Section 4.2.4. states the mathematical optimization problem formulization and the validity of applying weighted sum method in solving a multi-objective optimization problem. In this part, the weighted sum method is applied in the original delicatessen DFBN with VaR model between the DBNs and SD. The new DFBN that enables VaR and optimization is called the Optimized Dynamic Flow Bayesian Network (ODFBN). In order to present the delicatessen supply chain problem, several new variables are brought into the model for the ODFBN (Table 4.6). Applications of those variables are explained in the following problem formulization sections.

Table 4.6 New Variables and Their Attributes for the Sample Supply Chain.

Model of Variable	Variable Type	Variable Name	Variable Attribute
System Dynamics (SD)	Economics	Retailer's Selling Price for the Food (SP)	$SP^i = \text{ the selling price for the food at time } i$
		Retailer's Supply Chain Cost (SCC)	$SCC^i = \text{ retailer's supply chain cost at time } i$
		Retailer's Purchasing Price for the Food (RPP)	$RPP^i = \text{retailer's purchasing price of the food at time } i$
		Retailer's Inventory Capacity (RIC)	$RIC^i = \text{retailer's inventory capacity at time } i$
		Customer's Maximum Purchasing Rate (Demand) with Zero Selling Price (C)	$CPR^i = C - S \cdot SP^i, when\ SP^i \to 0, C = CPR^i$
		Slope of Price-demand Curve (S)	$S = \dfrac{C - CPR^i}{SP^i}$

i) Decision Variables and Objectives.

The decision variables for the optimization problem are SP^i, CPR^i and RPR^i. Smith (1973) states the multi-objective engineering economics optimization problem with the maximization the weighted sum of profitability and liquidity. Although, in some definitions, cash, accrued expenses, prepaid expenses and other factors affect the liquidity of a company, the requirement of working capital in supply chain strategy has a strong influence on the liquidity (Brandenburg, 2013). Guillén et al. (2005) apply working capital and profit as multi-objective optimization elements in supply chain risk analysis. Steuer & Na (2003) and Alfaro-Cid et al. (2008) also discuss the working capital application in multi-objective decision-making. Thus, working capital can be used to represent the liquidity position of the company with the neglection of other components in our case study.

Working capital of the delicatessen is the sum of inventory and trade receivables deducted by trade payables (Brealey et al., 2008). It can be denoted as:

$$Liquidity\ at\ time\ i = \left[RI^0 + \sum_{j=1}^{i}\left(RPR^j - CPR^j\right)\right]RPP + RSR^i SP^i - RPR^i SCC^i. \quad (4.51)$$

The profitability of the store is the product of net profit for each product sold and the sales rate (RSR^i). This relationship can be denoted as:

$$Profitability\ at\ time\ i = \left(SP^i - SCC^i\right)RSR^i. \quad (4.52)$$

Thus, by putting w_1 and w_2 as the weight of each factor, the objective function is:

$$Maximize\ \ w_1\left(SP^i - SCC^i\right)RSR^i + w_2\{\left[RI^0 + \sum_{j=1}^{i}\left(RPR^j - CPR^j\right)\right]RPP + RSR^i SP^i - RPR^i SCC^i\}. \quad (4.53)$$

ii) Price-demand Curve Constraint.

In the ODFBN, the variable CPR^i can be improved. In the original DFBN, CPR^i is determined by Equation 4.41. It means that if the Customer Order Level (COL^i) is high at time i, $COL^i = 100$; on the other hand, if the COL^i is low at time i, $COL^i = 50$. However, it is imprecise to conclude the impact value of risk. Comprising with SP^i, CPR^i can be calculated in the price-demand curve (Chopra & Meindl, 2010). The price-demand curve can be obtained through the parameter estimation of a down-sloping formula by using the past data. The estimated formula can be simplified as a linear equation as shown in Equation 3.9. This formula assumes that for a group of customers, the quantity demand for a specific merchandise (i.e. CPR^i) changes linearly with different prices (i.e. SP^i) (Equation 4.54).

$$CPR^i = C - S \cdot SP^i. \quad (4.54)$$

From a mathematical optimization problem point of view, the available range for CPR^i should be less than or equal to the right-hand-side of Equation 4.54. Thus, transformed from the price-demand curve equation, the price-demand curve constraint can be denoted as:

$$S \cdot SP^i + CPR^i \leq C. \tag{4.55}$$

Similar to other linear equations, constant C and the slope S can be gained by parameter estimation techniques, such as MLE, the method of moments, or simply two pairs of past CPR^i and SP^i values.

iii) Selling Price Constraint.

The selling price at any time slice i should larger than the supply chain cost. Otherwise, it is not possible for the store to be profitable:

$$SP^i \geq SCC^i. \tag{4.56}$$

iv) Retailer's Inventory Level Constraint.

The retailer's inventory level (RI^i) at any time slice i should not exceed the capacity of the inventory of the store (RIC^i):

$$RI^i \leq RIC^i. \tag{4.57}$$

The retailer's inventory level (RI^i) is the accumulated retailer's net purchasing rate ($\sum_{j=1}^{i} RPR^j - \sum_{j=1}^{i} CPR^j$) with the initial inventory level (RI^0):

$$RI^i = RI^0 + \sum_{j=1}^{i} RPR^j - \sum_{j=1}^{i} CPR^j. \tag{4.58}$$

Thus, by substituting RI^i in Equation 4.57 with the equalities in Equation 4.58, and transferring all decision variables to the LHS, Equation 4.57 can be reformulated as:

$$\sum_{j=1}^{i} RPR^j - \sum_{j=1}^{i} CPR^j \leq RIC^i - RI^0. \tag{4.59}$$

v) Purchasing Rate Constraint.

The customer's purchasing rate (CPR^i) should be greater than or equal to the customer's expected unfilled order rate ($CEUOR^i$):

$$CPR^i \geq CEUOR^i \tag{4.60}$$

vi) Retailer's Procurement Constraint.

In order to keep the freshness of the food and the enough inventory for customer satisfaction, the manager of the store decides that once the inventory level of the store (RI^i) is below 150 units, the store procures one batch of this kind of food. Thus, the retailer's procurement rate should be:

$$RPR^i = \begin{cases} RSR^i + 50, & RI^i \leq 150 \\ RSR^i, & Otherwise \end{cases}. \tag{4.61}$$

vii) All Decision Variables Should Be Real Numbers.

$$SP^i \geq 0, CPR^i \geq 0, RPR^i \geq 0. \tag{4.62}$$

By assuming the weight requirement for the profitability and liquidity as w_1 and w_2, the simulation result by MATLAB computation is shown in Figure 4.9. In Figure 4.9, several key operation statistics of the store can be investigated. First, the retailer's inventory level RI^i is retained around the required 150 units from the third month to the twelfth month (Figure 4.9a). Second, the recommended selling price SP^i for the food is generated for each month with the floating of customer's demand (Figure 4.9b). Third, the predicted retailer's profit by month with w_1 weight in the objective function (Figure 4.9c). Fourth, the retailer's liquidity denoted by the working capital by month with w_2 weight in our objective (Figure 4.9d). At last, Figure 4.9e illustrates the customer's and retailer's procurement rate in the next twelve months. From the figure, we can observe that variance of RPR^i is higher than CPR^i. The two lines in the figure indicate a Bullwhip

Effect, which is an amplification phenomenon of demand variability from a downstream entity (i.e. customer) to an upstream entity (i.e. retailer) (Torres & Morán, 2006).

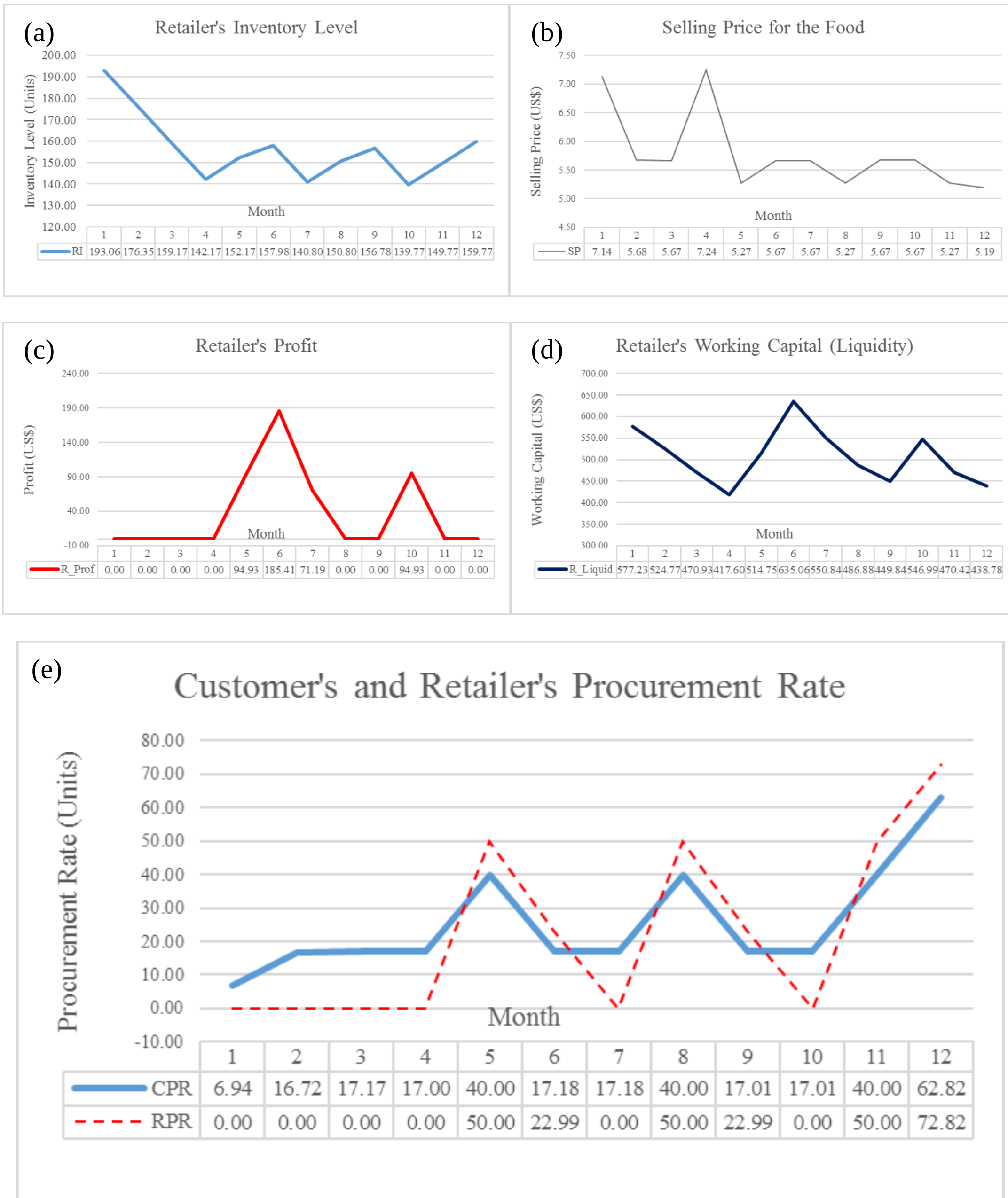


Figure 4.9 The Result of the ODFBN for the Delicatessen Case Study by time i: (a) RI^i; (b) SP^i; (c) Retailer's Profit; (d) Retailer's Working Capital; and (e) CPR^i and RPR^i.

Compared with the original DFBN, the revised ODFBN model has various improvements toward its functionality and the mathematical foundations. The following section disintegrates the hybrid model and analyzes the two models side by side.

4.3.5. Comparison of the DFBN with the ODFBN

The original DFBN model and the preliminary delicatessen case study result were published in the *Proceedings of the 2016 Industrial and Systems Engineering Research Conference* with the title "An Integrated Dynamic Flow Model for Supply Chain Risk Analysis" (Sun & Luxhøj, 2016). These three perspectives: problem identification, model formulation and result generation are presented in the conference paper. In this section, a comparison between the DFBN and the ODFBN is conducted from these three perspectives (Table 4.7).

Table 4.7 The Comparison between the DFBN and the ODFBN.

<table>
<tr><th></th><th>DFBN</th><th>ODFBN</th></tr>
<tr><td rowspan="2">Problem Identification</td><td colspan="2">• Identify the intended output variable</td></tr>
<tr><td>• Identify the optimization objectives through empirical derivation</td><td>• Identify the optimization objectives through quantitative models</td></tr>
<tr><td rowspan="2">Model Formulation</td><td colspan="2">• Formulate supply chain structure (SD)
• Identify risk factors and formulate causal relationships (DBN)
• Establish replenishment plans
• Obtain probability values in the DBN verbal-numerical mappings
• Predict evidence of risks and deploy mitigations</td></tr>
<tr><td>• Link DBN and SD through expected value method and decide the impact value from empirical derivation</td><td>• Link DBN and SD through VaR and generate risk distribution through parametric estimation from the past disruption data
• Estimate the price-demand curve and generate customer demand from the curve</td></tr>
<tr><td>Result Generation</td><td>• Yield various results by adjusting different replenishment plans
• Manually select preferred solution</td><td>• Automatically yield optimal solutions for formulated model</td></tr>
</table>

For both the DFBN and the ODFBN, the first is identifying the intended output variable and optimization objectives. However, the objectives of DFBN can only be concluded by people's experience. For the ODFBN, we can use various financial, economics, statistics or other quantitative models construct objective function. In the model formulation, the basic structure of both the DFBN and the ODFBN requires a supply chain structure, risk factors and causal relationships, replenishment plans, probability values by using verbal-numerical mappings and evidence and mitigations. The main difference between the DFBN and the ODFBN is the linkage model between DBN and SD. The DFBN uses the straightforward expected value method, which is the product of probability and its impact value (Equation 4.13). For the ODFBN, the VaR model is applied to connect DBN and SD. Parametric estimation methods can be used to determine the probability distribution by using the past supply chain risk or disruption data. Both problem identification and model formulation of the ODFBN are based on past data, which makes the model more referable and reliable for professional analysis. For a DFBN, the result of a single replenishment plan may not satisfy the requirement of the supply chain practitioner. Thus, various trials of different plans are needed to generate an approximately optimal solution. However, for the ODFBN, the optimal solution for a specific objective will be automatically obtained after the simulation. In the fast pace business environment, the ODFBN seems to be more suitable for industrial implementation.

In some industrial practices, a model that expresses the rivaly nature between different business entities may be better for application. However, there are not adequate arguments in the ODFBN model that show the competitive interaction between different stages of a system. Thus, in the next chapter, an EDFBN model is presented to satisfy the modeling needs in such scenario.

5. DFBN with Network Equilibrium

In international collaboration, information sharing in the business world makes commercial activities more evident. Not only the cooperation, but also the competition is more transparent and unconcealed. At this stage, the DFBN and the ODFBN show limited capability on reflecting the intercorporate interactions in a system. Thus, the concept of Supply Chain Network Equilibrium is introduced in this chapter to analyze the behaviors and optimal conditions among organizations.

5.1. Supply Chain Network Equilibrium

Supply Chain Network Equilibrium is originally introduced by Nagurney et al. (2002) based on the Nash (1950) equilibrium model. In the original presentation, a three-stage supply chain (manufacturers, retailers and demand markets) is analyzed with multiple business entities in each stage. The equilibrium is established on maximizing the profit for a company at a stage in a non-linear optimization problem. The optimality conditions, which are the constraints of the problem, are formed by each stage of the supply chain. The equilibrium for a manufacturer is achieved if the price that a retailer is willing to pay for a product is equal to the manufacturer's marginal transaction and production costs associated with the retailer. For a retailer, the equilibrium condition is: if consumers purchase products from a retailer, then the price charged by the retailer is the total cost of the product paid by the retailer. The equilibrium of the consumers' level considers both the price charged for the product and the transaction cost to obtain the product. The Supply Chain Network Equilibrium is the state that all of the product flows between different stages of the supply chain coincide and the prices and product flows satisfy the equilibrium conditions in each stage. The equilibrium condition for all stages of the supply chain is summarized by the variational inequality formulation. Nagurney et al. (2002) suggest a modified projection method of Korpelevich (1977) solve the variational inequality.

Based on the original Supply Chain Network Equilibrium by Nagurney et al. (2002), researchers develop distinct applications in the supply chain domain. Nagurney & Toyasaki (2005) and

Hammond & Beullens (2007) apply the model to the electric and electronic waste recycling problem in the European Union and Japan. Teng et al. (2007) expand the model to a multi-commodity flow model with random demand. Hamdouch (2011) develop a multi-period Supply Chain Network Equilibrium with purchasing strategies and traffic equilibrium. In the remaining research, we believe that the Supply Chain Network Equilibrium can be synchronized with a dynamic supply chain risk model such as the DFBN.

Inspired by the concept of Supply Chain Network Equilibrium, an Equilibrated Dynamic Flow Bayesian Network (EDFBN) is developed by modifying the ODFBN. The concentration of the EDFBN modeling is based on the multi-objective optimization with system profit maximization. The constraints of the optimization form the equilibrium conditions of the dynamic flows. The supply and demand at each stage of the system should reach its equilibrium state. The effects of the risks and mitigations can be propagated to the stocks and flows of the system. Insights and performance metrics based on the network equilibrium can be utilized for making strategic decisions. At the current stage, in an EDFBN, several assumptions should be defined:

1) time is discrete;

2) pricing, selling and demand information are shared in the modeled system concurrently within each time slice;

3) every business entity in the system is rational, strives for its own good and remains competitive to each other;

4) the production rate of the first stage of the system equals the quantity demanded of the market;

5) the ideal amount of the dynamic flow is obstructed by the delays or unfulfillment of orders that are caused by the potential disruptions.

Figure 5.1 demonstrates a sample EDFBN with a three-stage supply chain that includes m manufacturers, r retailers and n demand markets. The solid-line arrows show the dynamic flows between the business entities and the dash-line arrow illustrate the demand information flows.

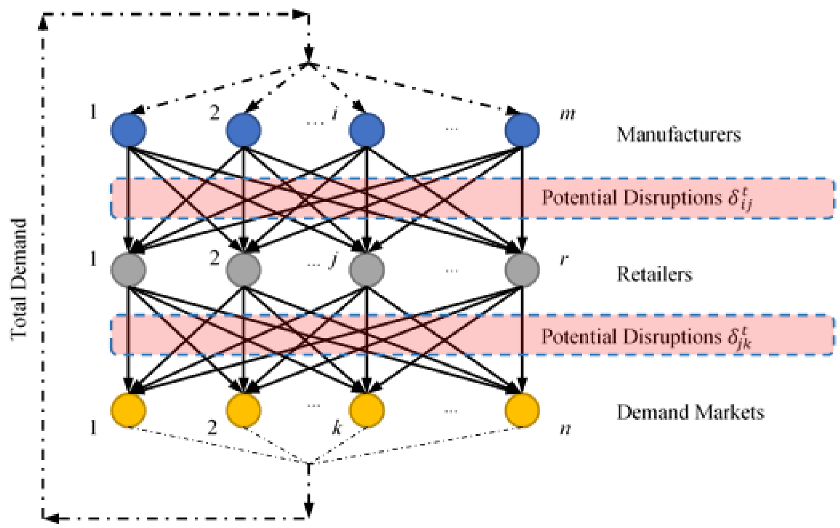


Figure 5.1 A Sample EDFBN for a Three-Stage Supply Chain.

Decision variables for this optimization are the flow variables: $Q^t = \{q_{ij}^t, q_{jk}^t\}$ and price variables $P^t = \{\rho_i^t, \rho_j^t, \rho_k^t\}$ for $i \in \{1, \dots, m\}, j \in \{1, \dots, r\}$, and $k \in \{1, \dots, n\}$, and for each time slice t in the total time T. Dynamic flow q_{ij}^t or q_{jk}^t may be affected by the elements in the set of flow variables Q_{ij}^t and Q_{jk}^t at time slice t. The RAVs in this EDFBN (δ_{ij}^t and δ_{jk}^t) propagate the temporal and systemic risk into the dynamic flows.

The objective functions, which aim to achieve the maximum profit for every single stage of the supply chain at each time slice t, are denoted in Table 5.1. Table 5.1 also includes the constraints of the optimization.

Table 5.1 The Objective Functions and The Constraints of the EDFBN.

Categories	*Descriptions*	*Mathematical Expressions*
Objective Functions	Maximizing manufacturers' profit	$maximize\ \sum_{i=1}^{m}\sum_{j=1}^{r}\rho_{ij}^{t}(q_{ij}^{t}-\delta_{ij}^{t}) - \sum_{i=1}^{m}\sum_{j=1}^{r}c_{ij}^{t}(Q_{ij}^{t},\delta_{ij}^{t})$
	Maximizing retailers' profit	$maximize\ \sum_{j=1}^{r}\sum_{k=1}^{n}\rho_{jk}^{t}(q_{jk}^{t}-\delta_{jk}^{t}) - \sum_{j=1}^{r}\sum_{k=1}^{n}c_{jk}^{t}(Q_{jk}^{t},\delta_{jk}^{t})$
	Minimizing demand markets' cost	$minimize\ \sum_{j=1}^{r}\sum_{k=1}^{n}\rho_{k}^{t}(q_{jk}^{t}-\ \delta_{jk}^{t})$
Flow Constraints	The sum of the manufacturer's productivity is less than or equal to the total market demand at the manufacturer's unit price willing to sell at	$\sum_{i=1}^{m}(q_{di}^{t}-\delta_{di}^{t}) \le D(\rho_{m}^{t})$
	The amount of product outflow of each retailer to all demand markets should be less than or equal to the amount of inflow from all manufacturers	$\sum_{k=1}^{n}(q_{jk}^{t}-\delta_{jk}^{t}) \le \sum_{i=1}^{m}\left(q_{ij}^{t}-\delta_{ij}^{t}\right), \forall j \in \{1,2,\dots,r\}$
	The sum of the amount of the retailer's product willing to sell should be greater than or equal to the total market demand at the manufacturer's unit price willing to sell at	$D(\rho_{m}^{t}) \le \sum_{k=1}^{n}\sum_{j=1}^{r}\left(q_{jk}^{t}-\delta_{jk}^{t}\right)$
Cost and Price Equilibrium Constraints	Each manufacturer's unit price willing to sell at is greater than or equal to the unit cost of goods sold	$c_{ij}^{t}(Q_{ij}^{t},\delta_{ij}^{t}) \le \rho_{i}^{t}$
	Each retailer's unit price willing to sell at is greater than or equal to the unit cost of goods sold	$c_{jk}^{t}(Q_{jk}^{t},\delta_{jk}^{t}) \le \rho_{j}^{t}$
	The unit price that each demand market is willing to pay is less than or equal to the retailer's unit price willing to sell	$\rho_{k}^{t} \le \rho_{j}^{t}$
Non-Negativity of the Actual Flows	Dynamic flows from manufacturers to retailers	$q_{ij}^{t}-\delta_{ij}^{t} \ge 0$
	Dynamic flows from retailers to demand markets	$q_{jk}^{t}-\delta_{jk}^{t} \ge 0$
Non-Negativity of the Decision Variables		$q_{ij}^{t}, q_{jk}^{t}, \rho_{i}^{t}, \rho_{j}^{t}, \rho_{k}^{t} \ge 0$

With the influence of the temporal risk at every time slice t, the optimization formulated shown in Table 5.1 is executed once. By performing the optimization T times, the fluctuation of the dynamic flows, inventory levels and the optimal value for the objective functions can be obtained. After introducing the basic features of the EDFBN, the problem of the modeling shifts to increase the effectiveness of the application. A Micro Supply Chain concept is proposed to change the conventional perspective of the model application.

5.2. Micro Supply Chains

Traditional supply chain research focuses on the inter-corporate flows of information, capital and physical materials. The globalized business cooperation nowadays leads to companies in one supply chain not only geographically, but also economically, demographically and politically separated. Nondisclosure agreements from each corporation also isolate the critical quality control information flowing between them. It is difficult for engineers and consultants to collect adequate data for systemic supply chain risk analysis in such a scenario. After consulting with industry practitioners, an operational level, Micro Supply Chain (MSC) concept is inspired.

An MSC describes the processes of transferring inputs to outputs in a single production site. In an MSC, similar with a traditional supply chain, dynamic flows of manufacturing materials, information, capital and people can be still discovered. Systemic risks can still be observed in the procedures of production, such as the failures of machines, workers, or coordination. Compared with an inter-corporate supply chain, the production data are easier to gather for an MSC. A single company's permission is needed to access the required data. In sum, in order to generate more realistic results for decision-making, we find the MSC is an ideal direction to expand the EDFBN model.

In addition, in the ODFBN, the weighted sum method is applied to obtain the optimal solutions of the multi-objective optimization. The weights assigned to the objectives may not ensure the most optimized value for the multi-objective problem. In the EDFBN, Genetic Algorithms (GA) are applied to solve the optimization problems.

5.3. Genetic Algorithms

Holland (1975) introduces a Genetic Algorithm (GA) by imitating the theory of evolution in nature. In nature, the natural selection process threats the weak and unfit species with the risk of extinction. On the contrary, a greater possibility of reproduction is given to the strong ones. As this effect accumulates, the species carrying the most suitable combinations of genes survive the environment. During the long process of evolution, there are some very improbable chances of mutation may occur in genes. The changes that provide an additional probability of survival help the evolution process of the species. On the other hand, natural selection eliminates the species with the inferior genes caused by the mutations (Konak et al., 2006). In GA, a chromosome is a solution vector. As the unit of evolution, genes form the chromosomes. A unique solution in the solution space is denoted by a chromosome. The projection of the chromosomes to the solution space is called encoding, which this the objective of GA. A collection of chromosomes is defined as a population. When the GA is operated, the population is normally randomly initialized. As the GA processes advance, the chromosomes converge into fitter ones, and eventually, the population is dominated by a single chromosome.

There are two operators in a GA to generate new solutions: crossover and mutation. Crossover combines two parent chromosomes together to produce new offspring chromosomes. The offspring prefers good genes from its parents, which makes the parents with fitter genes more preferred from their population. After applying the crossover iteratively, more chromosomes with preferred genes are expected in the population. Gradually, an optimal solution is converged from this process. Mutation is the process of randomly changing the characteristics of the chromosomes at the gene level. Mutation rarely occurs in the iterations of the GA computation. Even though the mutation occurs in low probability and affects the entire chromosome negligibly, the GA is not effective without the mutation. In a GA, crossover trends to uniform the characteristics of chromosomes. On the contrary, mutation occasionally counters the convergent effect of crossover and helps the GA avoid local optima.

Selection is the next step in GA after crossover and mutation. The criterion for selection is based on the fitness of the chromosomes to the environment. According to Konak et al. (2006), various selection procedures are developed for GA depending on how the fitness values are applied, such as proportional selection, tournament selection and ranking selection. If the fitness of the chromosomes satisfies the criterion, then the GA is stopped; otherwise, the next iteration of crossover and mutation continue to perform until the stopping threshold is passed.

Numerous researchers have adapted the GA to solve multi-objective optimization problems (Deb et al., 2000; Konak et al., 2006; Horn et al., 1994; Asadi et al., 2014). GAs possess the attribute of maintaining and searching for non-inferior solutions in a constant population. This characteristic makes a GA appropriate for solving multi-objective optimization problems (Fonseca & Fleming, 1993). After elaborating the basic ideas of the GA, the following section presents an MSC case study to thoroughly introduce the application of the EDFBNs in supply chains.

5.4. An EDFBN Case Study

A system of feedbacks, stocks and dynamic flows that are exposed to temporal and systemic risks construct a supply chain. The EDFBN is applied in the supply chains for accomplishing the supply/demand equilibrium and maximizing the profitability of different stages under the potential disruptions. In this section, an EDFBN is established to model and analyze an MSC case study that can provide insights at an operational-level system.

5.4.1. Problem Identification

In order to enhance the performance of a window assembly line of a mid-sized window manufacturer, the supply chain manager investigates the production system and illustrates it in Figure 5.2. The manager observes that some random risk events may cause production delays or disruptions to the production line. After summarizing his supply chain managerial experience, the manager identifies the key risk factors and their causal relationships in this MSC by referring to the SCRIPT. By modeling the systemic risks with the dynamic flows of the window assembly line, an EDFBN is established in Figure 5.3.

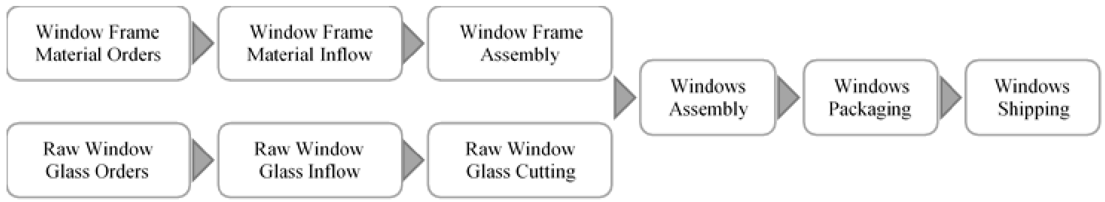


Figure 5.2 A Window Assembly Line of a Window Manufacturer.

5.4.2. EDFBN Modeling

In Figure 5.3, the DBNs are built with GeNIe (http://www.bayesfusion.com/), and the SD is modeled with Vensim PLE (http://vensim.com/). With all production processes reaching their equilibrium state, the optimization aims to maximize the total profit of the assembly line. We use MATLAB (https://www.mathworks.com/) as the data processing and mathematical modeling and execution tool. The time span for the EDFBN is $T = 12$ months with each time slice t representing 1 month.

5.4.2.1. Systemic Risks with Evidence and Mitigation

For each RAV (dynamic flow variables in blue characters in Figure 5.3), there is an OV that provides the risk information. For each OV, there is a DBN that propagates the original risk information to it. The arrows with a numbered text box between the OVs implies that the variables influence their descendant nodes in the next one time slice. For instance, the variable Glass Cutting Delay (GCD) is the OV of the RAV Window Glass Cutting Delay (WGCD). The GCD is affected by the Window Raw Glass Supplying Delay (WRGSD) from the previous time slice. The values in the CPTs of the DBNs are translated from the linguistic expressions of the risk occurrences to probability values by applying verbal-numerical mappings.

Similar to the DFBNs and the ODFBNs, some evidence of risk events can be identified in the DBNs of the EDFBNs. Mitigations can be deployed to counter the negative impact of the disruptions to the system when the risk events are observed. The mathematical mechanisms of the evidence and mitigation are introduced in Equations 4.17, 4.18, 4.21 and 4.22.

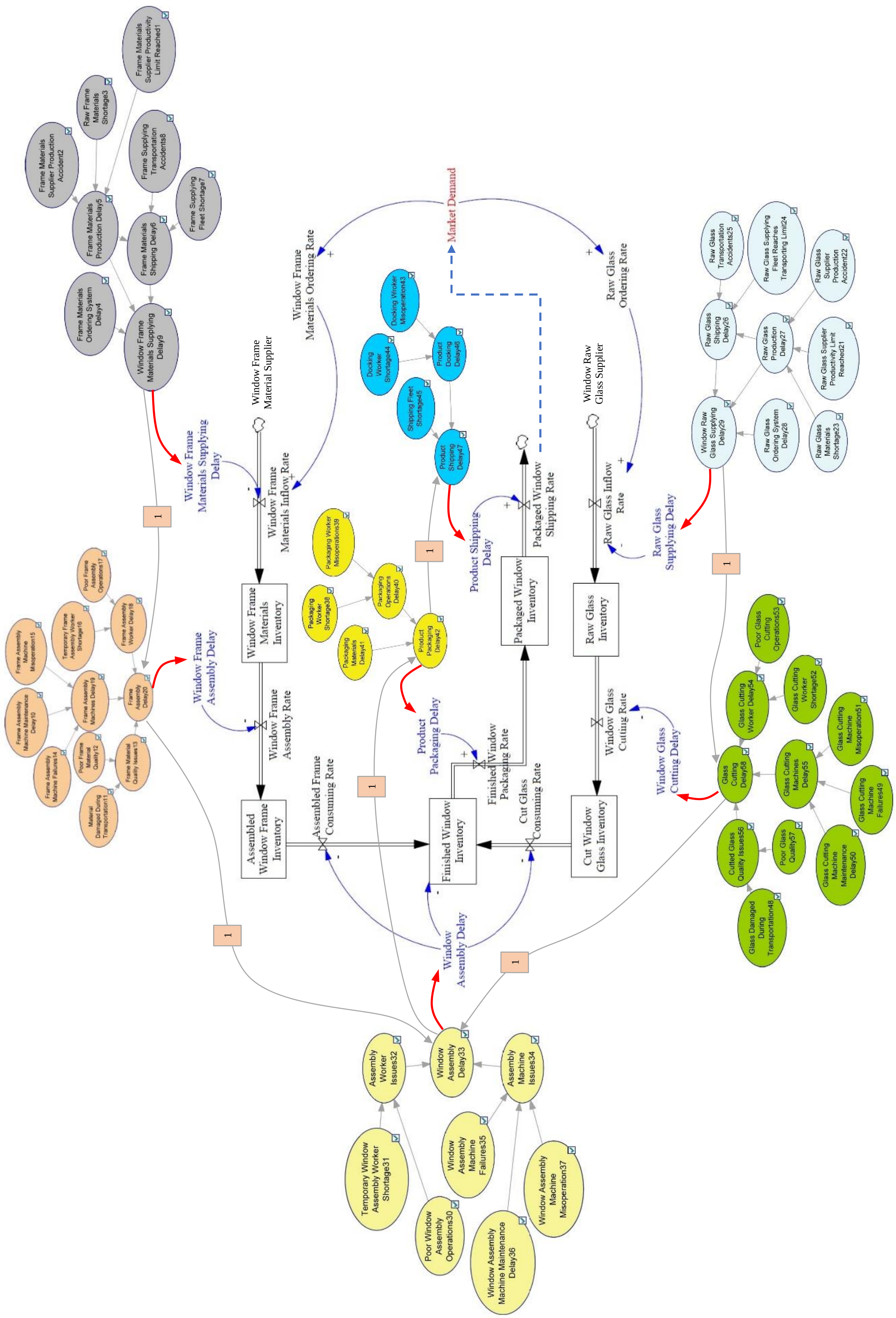


Figure 5.3 The EDFBN for the Window Assembly Line.

In this window assembly line case study, the observed evidence of the risk occurring is summarized in Table 5.2. In order to restrain the impact of the disruptions to the system, mitigations plans are also implemented and recorded in Table 5.2.

Table 5.2 Evidence and Mitigations for the Window Assembly Line.

Time (Month)	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Frame Material Supplier Production Accident	Yes	No								Yes	No	
Supplying Fleet Shortage			Yes				Yes					
Frame Assembly Machine Maintenance Delay	Yes	No		Yes	No		Yes	No		Yes	No	
Frame Assembly Machine Failure		Yes							Yes			
Raw Glass Materials Shortage					Yes							
Raw Glass Transportation Accidents			Yes			Yes						
Poor Window Assembly Operations	Yes		Yes		Yes	No						
Packaging Worker Shortage							Yes	Yes	Yes	No		
Docking Worker Shortage							Yes	Yes	Yes	No		
Glass Cutting Machine Misoperation			Yes						Yes			
Poor Glass Quality					Yes							

Evidence Mitigation

5.4.2.2. Inventory and Dynamic Flows Under Risks

The inventories in this case study are calculated as the difference between the accumulated inflow and outflow of the material. The rules of computation between the inventory level and dynamic flows of the MSC are denoted in Table 5.3.

Table 5.3 The Inventory Levels Calculated by the Dynamic Flows.

Inventory levels in the MSC	***Dynamic flows that affect the inventory level***	***Mathematical Expressions***
Window Frame Materials Inventory (WFMI)	Window Frame Materials Inflow Rate (WFMIR), Window Frame Assembly Rate (WFAR)	$WFMI^t = WFMI^0 + \sum_{\tau=1}^{t}(WFMIR^\tau - 4 * WFAR^\tau)$
Assembled Window Frame Inventory (AWFI)	Window Frame Assembly Rate (WFAR), Assembled Frame Consuming Rate (AFCR)	$AWFI^t = AWFI^0 + \sum_{\tau=1}^{t}(WFAR^\tau - AFCR^\tau)$
Raw Glass Inventory (RGI)	Raw Glass Inflow Rate (RGIR), Window Glass Cutting Rate (WGCR)	$RGI^t = RGI^0 + \sum_{\tau=1}^{t}(RGIR^\tau - \frac{1}{3} * WGCR^\tau)$
Cut Window Glass Inventory (CWGI)	Window Glass Cutting Rate (WGCR), Cut Glass Consuming Rate (CGCR)	$CWGI^t = CWGI^0 + \sum_{\tau=1}^{t}(WGCR^\tau - CGCR^\tau)$
Finished Window Inventory (FWI)	Assembled Frame Consuming Rate (AFCR), Cut Glass Consuming Rate (CGCR), Finished Window Packaging Rate (FWPR)	$FWI^t = FWI^0 + \sum_{\tau=1}^{t}(\min(AFCR^\tau, CGCR) - 10 * FWPR^\tau)$
Packaged Window Inventory (PWI)	Finished Window Packaging Rate (FWPR), Packaged Window Shipping Rate (PWSR)	$PWI^t = PWI^0 + \sum_{\tau=1}^{t}(FWPR^\tau - 10 * PWSR^\tau)$

In this case study, we assume that every dynamic flow is exposed to the temporal systemic risks modeled by the DBNs. The influence of the risks to the dynamic flows is reflected by the RAVs in the system. It is assumed that the RAVs affect the flows linearly. For example, the Window Frame

Assembly Rate (WFAR) and the Assembled Frame Consuming Rate (AFCR) at time t can be denoted by:

$$WFAR^t = WFAR^{t^*} - WFAD^t \tag{5.1}$$

$$AFCF^t = AFCR^{t^*} - WAD^t \tag{5.2}$$

where $WFAD^t$ and WAD^t are the Window Frame Assembly Delay and the Window Assembly Delay at time t; $WFAR^{t^*}$ and $AFCR^{t^*}$ are the ideal dynamic flow amounts without any delay at time t. With the inflow of WFAR and outflow of AFCR, the inventory level AWFI at time t can be expressed by:

$$AWFI^t = AWFI^0 + \sum_{\tau=1}^{t}[(WFAR^{t^*} - WFAD^t) - (AFCR^{t^*} - WAD^t)] \tag{5.3}$$

After constructing the SD of the window assembly line, the VaR that connects the SD and DBNs can be performed as introduced in Section 4.3.3.

5.4.2.3. Supply Chain Network Equilibrium Based Optimization

Multi-objective optimization in the EDFBN aims to predict the optimized material flows in supply/demand equilibrium state under the influence of risk factors of a complex system. After establishing the causal relationship of risks in DBNs, dynamic flows in SD and the linkage between those two models, the Supply Chain Network Equilibrium inspired multi-objective optimization is executed to generate both the integrated and independent behaviors of the different stages' decision-makers. The following part presents the formalization of the optimization.

At an arbitrary time slice t, we assume that the frame material supplier and the raw glass supplier are denoted by $i \in \{1,2\}$; the frame assembly process and the raw glass cutting process are denoted by $j \in \{a, b\}$; the windows assembly, packaging, shipping and demand market are represented by k, l, m and d, respectively; the decision variables for the optimization are price variables $P^t = \{\rho_{1a}^t, \rho_{2b}^t, \rho_m^t, \rho_d^t\}$ and quantity variables $Q^t = \{q_{d1}^t, q_{d2}^t, q_{1a}^t, q_{2b}^t, q_{ak}^t, q_{bk}^t q_{kl}^t, q_{lm}^t\}$. Table 5.4

presents the objective functions for the optimization. These functions focus on optimizing the profit or the cost of that stage of the supply chain.

Table 5.4 Objectives and Their Corresponded Objective Functions for the EDFBN.

Objectives	***RAVs that influence the objectives***	***Objective functions***
Maximize the suppliers' profit	Window Frame Materials Supplying Delay (WFMSD), Window Raw Glass Supplying Delay (WRGSD)	$maximize\ \rho_{1a}^{t}(q_{d1}^{t^*} - WFMSD^t) + \rho_{2b}^{t}(q_{d2}^{t^*} - WRGSD^t) - \{[c_{11}(q_{d1}^{t^*} - WFMSD^t)^2 + c_{12}(q_{d1}^{t^*} - WFMSD^t)(q_{d2}^{t^*} - WRGSD^t) + c_{13}(q_{d1}^{t^*} - WFMSD^t)] + [c_{21}(q_{d2}^{t^*} - WRGSD^t)^2 + c_{22}(q_{d2}^{t^*} - WRGSD^t)(q_{d1}^{t^*} - WFMSD^t) + c_{23}(q_{d2}^{t^*} - WRGSD^t)]\}$
Minimize the cost of raw material processing	Frame Assembly Delay (FAD), Glass Cutting Delay (GCD)	$minimize\ [c_{31}(q_{11}^{t^*} - FAD^t)^2 + c_{32}(q_{11}^{t^*} - FAD^t) + c_{33}](q_{11}^{t^*} - FAD^t) + [c_{41}(q_{22}^{t^*} - GCD^t)^2 + c_{42}(q_{22}^{t^*} - GCD^t) + c_{43}](q_{22}^{t^*} - GCD^t)$
Minimize the cost of window assembly process	Window Assembly Delay (WAD)	$minimize\ [c_{51}min(q_{1k}^{t^*} - WAD^t, q_{2k}^{t^*} - WAD^t)^2 + c_{52}min(q_{1k}^{t^*} - WAD^t, q_{2k}^{t^*} - WAD^t) + c_{53}] \cdot min(q_{1k}^{t^*} - WAD^t, q_{2k}^{t^*} - WAD^t)$
Minimize the cost of window packaging process	Product Packaging Delay (PPD)	$minimize\ [c_{61}(q_{kl}^{t^*} - PPD^t)^2 + c_{62}(q_{kl}^{t^*} - PPD^t) + c_{63}](q_{kl}^{t^*} - PPD^t)$
Minimize the cost of window shipping process	Product Shipping Delay (PSD)	$minimize\ [c_{71}(q_{lm}^{t^*} - PSD^t)^2 + c_{72}(q_{lm}^{t^*} - PSD^t) + c_{73}](q_{lm}^{t^*} - PSD^t)$
Maximize the profit of the window assembly line	Product Shipping Delay (PSD)	$maximize\ \rho_m^t(q_{lm}^{t^*} - PSD^t)$
Minimize the cost of the demand market	Product Shipping Delay (PSD)	$minimize\ \rho_d^t(q_{lm}^{t^*} - PSD^t)$

It is assumed that the quantity demanded for the windows from this window production line is linearly related to the window's unit price:

$$D(\rho) = d^0 - s \cdot \rho \tag{5.4}$$

where d^0 is the quantity demanded when the unit price drops to 0, and s is the slope of the demand curve. The parameters of Equation 5.4 can be obtained through MLE by using past selling data.

The constraints of the EDFBN optimization are listed in Table 5.5. The supply/demand equilibrium, direction and non-negativity of the dynamic flows are ensured by the constraints.

Table 5.5 Constraints of the EDFBN Optimization.

Constraint descriptions	***RAVs that influence the constraints***	***Constraint Expressions***
Finished product shipping flow should be larger than or equals to the quantity demanded at the manufacturer's preferred selling price	Product Shipping Delay (PSD)	$D(\rho_m^t) \leq 100(q_{lm}^{t^*} - PSD^t)$
The window frame material quantity demanded at the manufacturer's preferred selling price should be larger than or equal to the ordering rate	Window Frame Materials Supplying Delay (WFMSD)	$q_{d1}^{t^*} - WFMSD^t \leq 4D(\rho_m^t)$
The window raw glass quantity demanded at the manufacturer's preferred selling price should be larger than or equal to the ordering rate	Window Raw Glass Supplying Delay (WRGSD)	$q_{d2}^{t^*} - WRGSD^t \leq \frac{1}{3}D(\rho_m^t)$
The window frame material inflow should be larger than or equal to the frame material consumption rate	Frame Assembly Delay (FAD), Window Assembly Delay (WAD)	$q_{1k}^{t^*} - WAD^t \leq \frac{1}{4}(q_{1a}^{t^*} - FAD^t)$
The window raw glass inflow should be larger than or equal to the raw glass consumption rate	Glass Cutting Delay (GCD), Window Assembly Delay (WAD)	$q_{2k}^{t^*} - WAD^t \leq 3(q_{2b}^{t^*} - GCD^t)$
The assembled frame consumption rate should be larger or equal to the window assembly rate	Window Assembly Delay (WAD), Product Packaging Delay (PPD)	$q_{kl}^{t^*} - PPD^t \leq 10(q_{ak}^{t^*} - WAD^t)$
The cut glass consumption rate should be larger or equal to the window assembly rate	Window Assembly Delay (WAD), Product Packaging Delay (PPD)	$q_{kl}^{t^*} - PPD^t \leq 10(q_{bk}^{t^*} - WAD^t)$
The product packaging rate should be larger or equal to the product shipping rate	Product Packaging Delay (PPD), Product Shipping Delay (PSD)	$q_{lm}^{t^*} - PSD^t \leq 10(q_{kl}^{t^*} - PPD^t)$
The manufacturer's preferred selling price should be larger or equal to the sum of all involved during production		$\rho_{1a}^t + \rho_{2b}^t + \sum_{x=3}^{7}\sum_{y=1}^{3} c_{xy} \leq \rho_m^t$
The demand market's preferred purchasing price should be less than or equal to the manufacturer's preferred selling price		$\rho_d^t \leq \rho_m^t$
Non-negativity of actual material flows	Window Frame Materials Supplying Delay (WFMSD), Window Raw Glass Supplying Delay (WRGSD), Frame Assembly Delay (FAD), Glass Cutting Delay (GCD), Window Assembly Delay (WAD), Product Packaging Delay (PPD), Product Shipping Delay (PSD)	$(q_{d1}^{t^*} - WFMSD^t), (q_{d2}^{t^*} - WRGSD^t),$ $(q_{1a}^{t^*} - FAD^t), (q_{2b}^{t^*} - GCD^t),$ $(q_{1k}^{t^*} - WAD^t), (q_{2k}^{t^*} - WAD^t),$ $(q_{kl}^{t^*} - PPD^t), (q_{lm}^{t^*} - PSD^t) \geq 0$
Non-negativity of all variables		$Q^t = \{q_{d1}^t, q_{d2}^t, q_{1a}^t, q_{2b}^t, q_{ak}^t, q_{bk}^t q_{kl}^t, q_{lm}^t\} \geq 0$ $P^t = \{\rho_{1a}^t, \rho_{2b}^t, \rho_m^t, \rho_d^t\} \geq 0$

For every time slice, GA is conducted to obtain the multi-objective optimization solutions (gamultiobj() MATLAB function), and 200 points on the Pareto front are generated. Figure 5.4 and Figure 5.5 illustrates the results of the optimization corresponding to the dynamic flows and inventory levels, respectively. 200 points on the Pareto front, i.e. the optimal solutions, are presented as 200 lines of the flow rates and inventory levels by each time slice. The abruptly rising or declining flow rates demonstrate the fluctuation of the dynamic flows in the MSC. The upheaval of the dynamic flows propagates to the inventory levels. This effect is displayed in Figure 5.5.

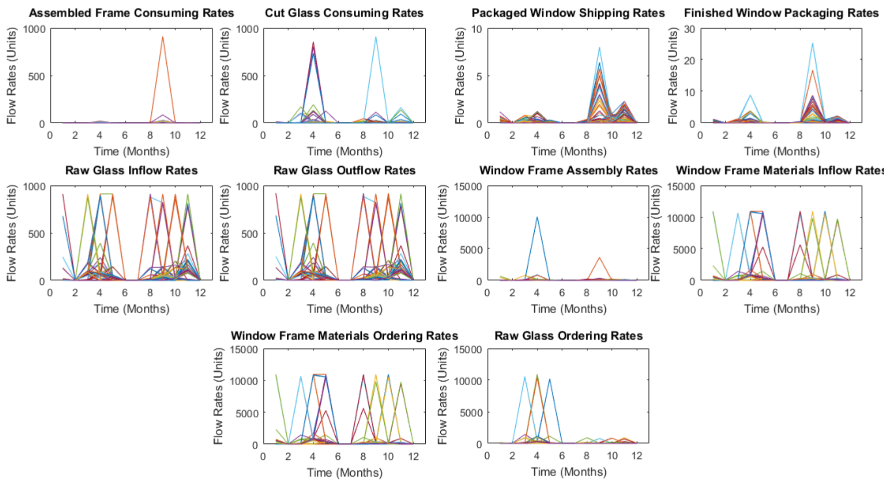


Figure 5.4 Optimization Results for the Material Flows.

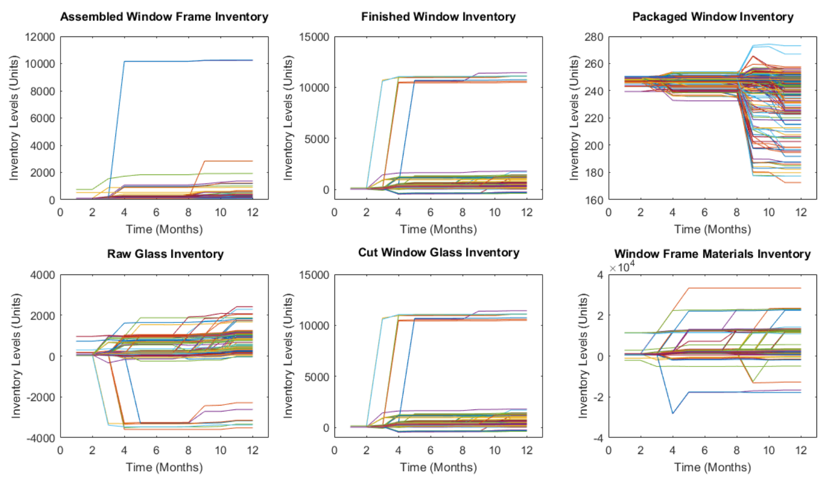


Figure 5.5 Optimization Results for the Inventory Levels.

5.4.3. EDFBN Case Modeling Results and Analysis

Figures 5.4 - 5.5 illustrates the influence of the temporal and systemic risks to the dynamic flows and inventory levels of the window assembly line. The supply chain manager can use the optimized window assembly line results for the next 12 months for decision-making, such as mitigation plans evaluation, pricing adjustment, production line optimization, etc. Additional supply chain performance metrics can also be provided to enhance the realistic implications of the results. This section also provides the validation of applying the network equilibrium concept to the EDFBN model.

- Monthly Fill Rate

In this case study, the delay of material delivery is defined as the phenomenon of the MSC's risks and disruptions. These effects are shown in the fluctuation of the dynamic flows in Figure 5.4. The fluctuations accumulate and amplify into the inventory levels in Figure 5.5. The sudden grow or drop of inventory levels may cause overstock or backorder of the MSC. The rocketing of inventory sharply increases the holding costs, while the shrinking of the stock rises the unfilled order rate. Disruptions like these have a deleterious the profitability and satisfactory of the supply chain. Thus, various supply chain metrics should be utilized to assess the performance of the supply chain under such influence.

Fill rate is one of the key metrics that can be obtained from the optimization results. It evaluates the on-time delivery rate of the ordered products. The window assembly line's monthly fill rate can be obtained from Equation 5.5.

$$Monthly\ Fill\ Rate = \left(1 - \frac{Monthly\ Total\ Order\ Amount - Monthly\ Shipping\ Amount}{Monthly\ Total\ Order\ Amount}\right) * 100\% \quad (5.5)$$

The average monthly fill rates for both with and without the mitigations listed in Table 5.2 for the 200 points on the Pareto front are generated to investigate the functionality of the mitigation plans. The average fill rates are shown in Figure 5.6 with 95% confidence intervals (CIs) for both scenarios.

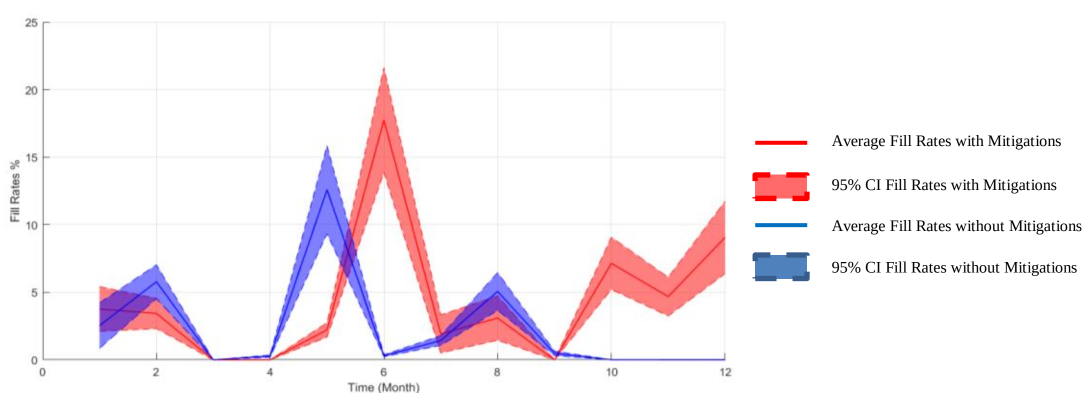


Figure 5.6 Comparison between the Average Fill Rates with and without Mitigations for the Window Assembly Line with 95% CI.

From the alterations of the fill rates, we can observe that except for the second, fifth and eighth month, the window assembly line with mitigations has higher fill rates than the production without the mitigations. It indicates that the manager can examine other mitigation plans by repeatedly applying the established EDFBN until the most effective and efficient mitigation plans are generated.

- Validity of Network Equilibrium

A major modification of the EDFBN is applying the network equilibrium concept, which maximizes the profitability for all stages of a system. Compared with the other two DFBN models without the network equilibrium, the EDFBN provides a more optimized system from the aspect of the microeconomics with preferable results.

For instance, in the window assembly line case study, the EDFBN is applied to generate the optimal dynamic flows with hidden risks in the MSC. At the same time, the optimal value, which is the profit of all parties in the MSC, is obtained in Figure 5.7 with a dash line. If the EDFBN is only modeled without the constraints corresponding to the Supply Chain Network Equilibrium concept, the optimal value of the optimization is obtained and illustrated in Figure 5.7 with a solid line.

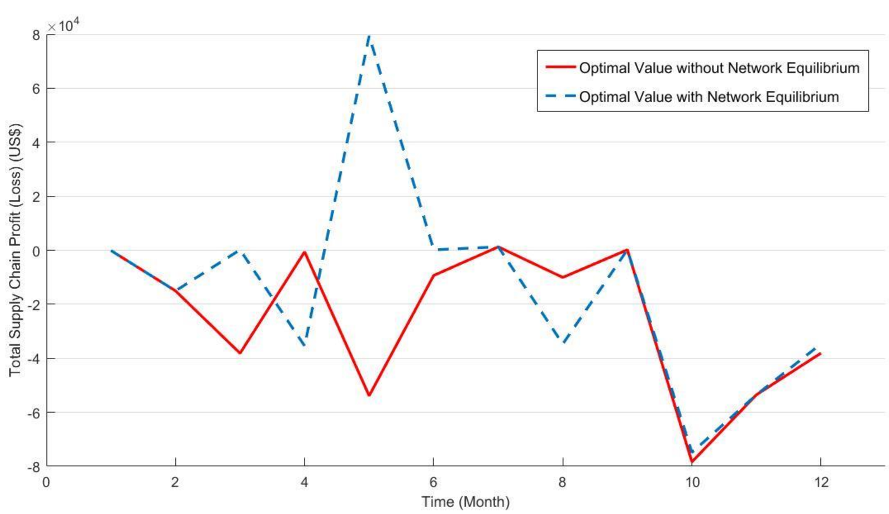


Figure 5.7 Comparison between the Optimal Value with and without the Application of Network Equilibrium.

As we can observe from Figure 5.7, the MSC yields better results, especially in the fifth month. The MSC only has an obvious disadvantaged profitability in the fourth and the eighth month. For the entire twelve months of simulation, the MSC *with* the network equilibrium has a loss of $16,721. However, a loss of $29,568 is projected for the system *without* the network equilibrium.

To summarize, with the major modification of the EDFBN, the network equilibrium aims at optimizing the profit objective for each stage of the supply chain that leads to the optimal total supply chain profit. The application of the EDFBN provides supply chain practitioners and other domains a valid tool that deeply optimizes their systems, which may obtain more desirable results for decision-making.

In the last chapter, a summary of the dissertation is presented with an evolutionary view of the DFBN models. A comparison between the DFBN models with other recent hybrid supply chain risk analysis model is also provided. Contributions of the research, current limitations and future directions of the research are also presented.

6. Conclusions and Future Research

This dissertation focuses on developing a series of novel risk modeling and analysis tools that are based on the theories of Bayesian Belief Networks, System Dynamics, Value-at-Risk, Genetic Algorithms, Mathematical Optimization and Supply Chain Network Equilibrium. This series of tools, termed by the DFBN models, provides a new perspective for supply chain analysis in a temporal and systemic risks environment where the functionality of dynamic flows in the system can be disrupted.

6.1. Methodology Summary

Supply chain potential disruptions in the dynamic business flows can be caused by various risk factors. By integrating SD and DBN, the DFBN model simulates the stocks and flows fluctuation with systemic risks. The simulated results can be used to investigate the performance of a supply chain. Mitigations and alternative replenishment plans can be discovered to satisfy the business objective of the manager. Based on the original DFBN, the ODFBN is introduced to improve the effectiveness of the model. VaR substitutes the expected value method as the new linkage between the probabilistic risk output of the DBN. Besides, the customer demand prediction extricates from the empirical inference with the application of the price-demand curve. Both the distribution function in VaR and the price-demand curve are obtained from parametric estimation. It helps the model to conform with the past business environment. Moreover, the ODFBN integrates multi-objective optimization in the whole simulation process. The profitability and liquidity of the business can be maximized with weights assigned to the sub-objectives. By applying the concept of network equilibrium and GA, the EDFBN is developed to discuss total supply chain profit maximization from every stage of the supply chain. Once the EDFBN is executed, the objectives of the problem are optimized to the optimal value with the supply/demand of every stage of the system equilibrated.

6.2. DFBN, ODFBN and EDFBN

The DFBN establishes the foundation of modeling and analyzing the dynamic flows under threats from probabilistic risks and disruptions. The ODFBN and EDFBN are the modified methods that widen the application range of the original model.

Table 6.1 summarizes the modifications and improvements of the ODFBN and the EDFBN that differ from the original DFBN.

Table 6.1 Comparison Among the DFBN, the ODFBN and the EDFBN.

	DFBN	**ODFBN**	**EDFBN**
Problem Identification	Identify and formulate optimization objectives through empirical derivation	Identify and formulate optimization objectives through quantitative models	Identify and formulate optimization objectives through quantitative models
Model Formulation	Link DBN and SD through expected value method and decide the impact value from empirical derivation	Link DBN and SD through VaR and generate risk distribution through parametric estimation from the past disruption data	Link DBN and SD through VaR and generate risk distribution through parametric estimation from the past disruption data
	Generate customer demand from expected value method	Estimate the price-demand curve and generate customer demand from the curve	Estimate the price-demand curve and generate customer demand from the curve
Result Generation	Manually select preferred solutions	Automatically yield optimal solutions for the weighted sum objectives and constraints	Automatically yield optimal solutions with equilibriums in all stages of a system
	No optimal values	Manually adjust the weights for different objectives	Calculate the most optimized value through GA
	Reach the optimal state with manual manipulation of dynamic flow rules	Reach the optimal state with the constraints of industrial capacities	Reach the optimal state with the constraints of dynamic flows and network equilibrium

To summarize, the SD and DBN are integrated by the DFBN that considers temporal and systemic risks into dynamic flow simulation (Sun & Luxhøj, 2016). The preferred system outcome can be manually selected from different solutions by adjusting distinct dynamic flow rules. The ODFBN is developed from the DFBN with the modifications on multi-objective optimization, Value-at-Risk and pricing segment strategy (Sun & Luxhøj, 2017). By applying the weighted sum method, the ODFBN focuses on obtaining the optimal solution for a multi-objective system. Variables are constrained with dynamic rules or limited capacity of the system. Based on the DFBN and the ODFBN, the modeling of EDFBN integrates the concept of network equilibrium. In an EDFBN modeled system, not only the intended objectives, but also the supply/demand equilibrium of every stage of a system can be satisfied. The network equilibrium ensures that the optimal state of the system fulfills the benefits and interests of all elements in the model. In addition, GA is applied in EDFBN to achieve the optimal solutions of an optimization problem. Compared with the weighted sum method, GA guarantees the optimal value reaching the most optimized answer for a number of objectives.

6.3. Comparison of Hybrid Temporal Belief Network Method with Other Approaches

In recent years, the globalization of supply chain businesses stimulates numerous novel discussions and analysis models on maximizing supply chain profit under risks and disruptions. The DFBN and the improved ODFBN and EDFBN are introduced to facilitate supply chain decision-making under temporal and systemic risks. Other researchers also show interest in this area and fabricate remarkable thought on supply chain models. This section presents and compares two latest models on quantitative supply chain risk analysis with our model to show the strengths and weaknesses for possible future improvement.

6.3.1. Sahay's Iterative Simulation-Optimization Model

Sahay (2016) introduces an iterative simulation-optimization hybrid model for chemical production supply chain analysis in his Ph.D. dissertation. The objective of the hybrid model aims to minimize the total cost with the material and information flows considered in a supply chain. The simulation model transfers inventory targets to the optimization model. In reverse, after the optimization, emission and shipment targets are delivered to the simulation (Figure 6.1). The iteration advances until the difference between the cost obtained from the simulation and the optimization is less than ε (Figure 6.2). After the goal is achieved through the simulation-optimization iteration, several other variables in the model can be extracted to provide insights for the supply chain. The variables include: transportation cost, inventory cost, backorder cost, production cost, etc.

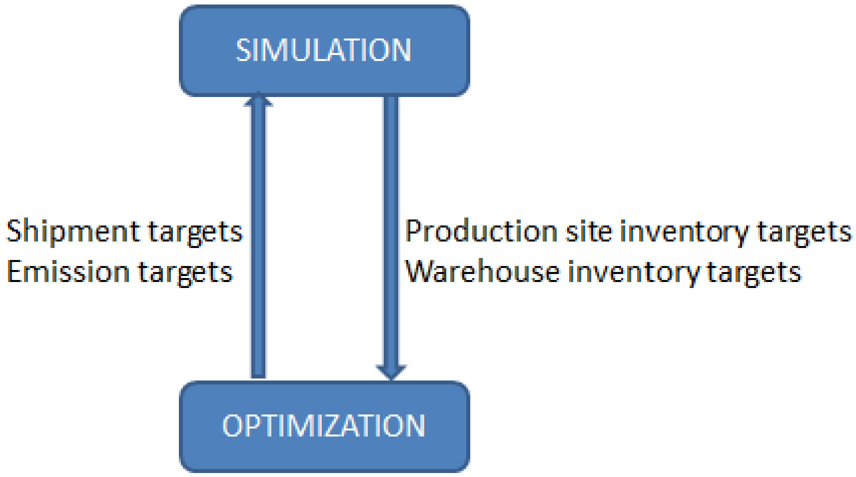


Figure 6.1 Iteration between Simulation and Optimization. (Source: Sahay, 2016)

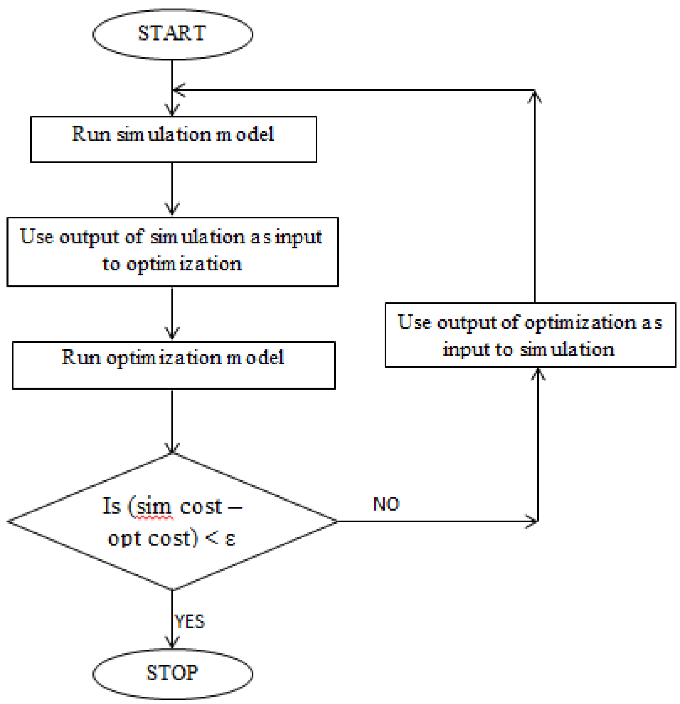


Figure 6.2 Looping Process of Simulation and Optimization. (Source: Sahay, 2016)

Sahay (2016) proposes the hybrid model with various supply chain application scenarios. First, centralized and decentralized supply chains are discussed in the dissertation. A four-stage supply chain is modeled with a centralized and a decentralized case. The cost comparison from each case is illustrated to assist the decision-making on selecting different supply chain strategy. Second, asynchronous and synchronous supply chain decision-making scenarios are implemented. In the synchronous scenario, information flows between supply chain entities simultaneously. The operation of an entity is synchronized with other sites. For the asynchronous scenario, the information from different entities is not instantly updated with each other. The operations of different entities are independent. The product shipping rates are generated from each scenario for decision-making. Third, a demand-uncertain case is introduced. A rolling horizon approach in a three-stage stochastic linear programming is conducted in the optimization model. Fourth, a derivative free optimization method is applied to the case that the analytic derivatives of the objective function are unavailable. A surrogate model is one of the derivative free optimization methods that employ surrogate approximations to overcome the high computational cost issue. At last, the simulation-optimization model is applied in a multi-enterprise supply chain that the buying and selling activities between retailers and warehouses are following auction mechanisms. The trend of bidding price is investigated with the altering in target inventory and learning rate.

In sum, the simulation-optimization model that Sahay (2016) introduces provides an iterative methodology that considers the dynamic flows of a supply chain. The dissertation shows the flexibility of the model in various supply chain application scenarios. The results decompose the costs, prices, profits and other indices that supply chain practitioners can obtain insights of a supply chain for assisting their decision-making.

6.3.2. Hahn & Kuhn's Robust Optimization Model

Hahn & Kuhn (2011) publish a value-based supply chain risk management tool with utilizing a robust optimization approach. The motivations of introducing this model can be mainly summarized in the following two points. (a) Past value driver tree models lack the ability to provide decision support; (b) only indirectly omitted scenario-based information can be obtained from risk implications for robust supply chain plans. Based on these motivations, the model focuses on generating direct results that reflect supply chain performance for decision-making by using robust optimization with various dynamic flows considered. Economic value added (EVA) is assigned as the metric of value-based supply chain performance. The optimization objective assumes that the decision-makers are risk-averse. The objective function is based on the upside potential (UP), downside risk (DR) and expected value (EV). According to Hahn & Kuhn (2011), the objective function can be denoted as:

$$\Phi(\gamma) = \gamma UP - DR = \gamma(UP - DR) - (1-\gamma)DR = \gamma EV - (1-\gamma)DR, \tag{6.1}$$

where $\gamma \in (0,1]$. The EV in Equation 4.63 is represented by the sum of EVA from various scenarios. The DR in Equation 6.1 is defined as the sum of the negative fraction of EVA under the scenarios. Constraints in the optimization problem are subject to the physical material and capital flows. Time value of money is considered in the capital flows.

A case study is presented with a four-stage supply chain. Physical materials flow from two suppliers to two plants. Then, two warehouses store finished products and deliver the products to five retailers. The supply chain involves three raw materials and three products. The customer demand is defined to be fluctuating seasonally with a probabilistic scenario model. The stochastic process is assumed to follow a triangular distribution. After the optimization is executed, a sensitivity analysis on four parameters (maximum overtime, the bank line of credit, the amplitude of demand seasonality and the hurdle rate) is conducted. The expected value of perfect information is utilized to illustrate the trade-off between the solution and objective robustness. The variable γ is adjusted to indicate the risk preference of a decision-maker (Figure 6.3). In addition, the ex ante and ex post information robustness regarding with the size of the scenario set is analyzed. It applies the discretization method in scenario sets of uneven size (Klugman et al., 2004).

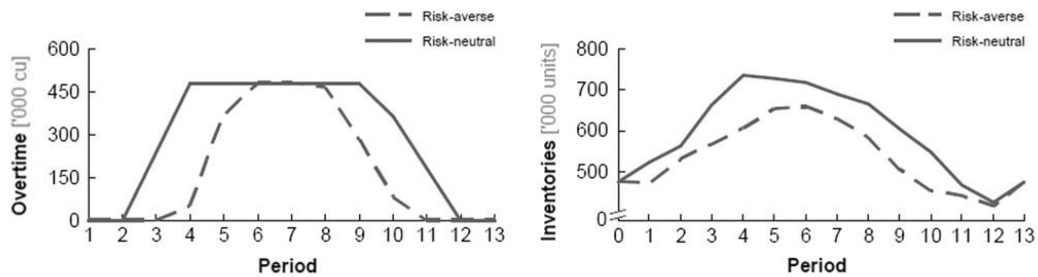


Figure 6.3 Overtime and Inventories for Risk-averse and Risk-neutral Decision-making Strategies. (Source: Hahn & Kuhn, 2011)

In Hahn & Kuhn's (2011) research, a decision-oriented value-based performance supply chain risk analysis model is proposed. The variables that leverage the performance and the source of supply chain operational risk are investigated. Various criteria of robustness are embedded in the model for achieving a comprehensive robust optimization. Abundant inventory and financial constraints ensure the rigorous application of the model.

In the following section, a side by side comparison is presented among three models: ODFBN, a simulation-optimization model of Sahay (2016) and a value-based robust optimization by Hahn & Kuhn (2011).

6.3.3. A Comparison Among the Models

A comparison among Sahay (2016), Hahn & Kuhn (2011) and the DFBN models is summarized in Table 6.2. The DFBN models include the original DFBN and the modified ODFBN and EDFBN. In the comparison, the strengths and weaknesses of our proposed DFBN models can be evaluated. From the comparison in Table 6.2, it can be observed that the DFBN models have advantages in the risk measurement method. The DFBN models utilize DBNs to create a temporal and systemic causal relationship of risks. It allows practitioners to disintegrate complex risk factors in a supply chain. For the other two models, downside risk model is applied. This method can only measure the overall risk that affects the economic value added. It lacks the ability to analyze supply chain risks from various sources. The DFBN models have several directions of development that can be discovered from the comparison. First, more prevalent economics and financial theories can be appended to the DFBN models. It can enhance the effectiveness of the model when more practical financial models are applied. Second, the application scenario of the DFBN models is limited to a three-stage supply chain and a six-stage supply chain at the current study phase. A more comprehensive supply chain model can be presented to show the compatibility, expandability and flexibility of the ODFBN in practical use.

Table 6.2 The Comparison Among the DFBN Models, Sahay (2016) and Hahn & Kuhn (2011).

Criteria	***The DFBN Models***	***Sahay (2016)***	***Hahn & Kuhn (2011)***
Modeling Objective	Multi-objective Optimization	Optimizing Cost	Value-based Performance
Core Quantitative Methods	System Dynamics and Dynamic Bayesian Network	Mixed Integer Linear Programming and Agent-based Simulation	Robust Optimization
Complementary Quantitative Methods	Nonlinear Programming, Value-at-Risk, Price-demand Curve, Maximum Likelihood Estimation, Genetic Algorithms, Network Equilibrium	Sensitivity Analysis, Design of Experiment, Mean Squared Error, Goal Seeking Approach, Rolling Horizon Approach, Expected Improvement, G-means	Sensitivity Analysis, Distribution Discretization
Risk Measurement Method	Dynamic Bayesian Networks	Downside Risk Model	Downside Risk Model
Inventory Theories Applied	Periodic Review Systems	Inventory Costs	Inventory Capacity, Production Capacity
Financial Theories Applied	Price-demand Curve, Working Capital, Supply/Demand Equilibrium	Multi-attribute Double Auction	Economic Value Added, Net Operating Profit After Tax, Net Operating Assets, Total Contribution Margin, Cash Flows, Account Receivable, Account Payable, Long-term Debts
Applied Scenarios	A Three-Stage Supply Chain, A Six-Stage, Two-Material, One-Product Micro Supply Chain	Centralized and Decentralized Supply Chain, Synchronous and Asynchronous Decision-making Supply Chain, Supply Chain Flexibility Assessment, Multi-enterprise Supply Chain Operation	A Four-Stage, Three-Material, Three-Product Supply Chain

6.4. Contributions of the Research

In sum, the main contributions of the research can be concluded in the following four aspects:

- **The Hybrid Approach of Integrating the DBN with the SD (A Major Contribution)**

The hybrid approach proposed, i.e. the DFBN models, integrates the risk analysis tool, the DBN, with the simulation method, the SD. The DFBN models provide a novel methodology for researchers and practitioners for analyzing the dynamic flows with temporal and systemic risks concurrently for a complex system. Executable MATLAB programs are also provided (Appendices I – III).

- **The Application of the Hybrid Approach to the Supply Chain Domain for Optimizing the Business Performance (A Major Contribution)**

The proposed DFBN models are applied to the supply chain with various domain-related methodologies, such as replenishment planning, price-demand curves, mathematical optimization, MSC, GA, multi-material, multi-stage supply chain and Supply Chain Network Equilibrium.

- **Connecting the Hybrid Approach with VaR Model (A Minor Contribution)**

The DFBN integrates the VaR as a complementary risk analysis model to connect the DBN and the SD. It provides a more satisfactory and controllable tool to support the linkage (RAVs) of the two crucial parts of the DFBN, i.e. the risk assessment (DBNs) and the flow simulation (SD).

- **Modeling the Hybrid Approach with the Concept of Network Equilibrium (A Minor Contribution)**

The EDFBN introduces the original dynamic flow risk analysis method with the concept of network equilibrium. By integrating with the network equilibrium concept, the EDFBN reflects a more realistic business circumstance that business entities remain collaborative and competitive at the same time. It ensures the optimization results satisfy the supply/demand equilibrium of every stage of a system.

6.5. Current Limitations and Future Research

The DFBN models are developed to unveil the influence of temporal and systemic risks to business systems with dynamic flows. Preliminary applications of the models are presented with two case studies. The expandability and flexibility have been proved with the elaboration of the modeling and results analysis. However, limitations of this research exist and there are opportunities for future improvement.

First, the linkage between the main methodologies SD and DBN of the DFBN models shows an interesting topic for future research. It is initially developed by using the expected value method in the DFBN. Then, VaR is applied as the linkage in the ODFBN and EDFBN. The GEVD applied in the VaR requires adequate data for parameter estimation. It may restrict the accuracy of model implementation in a data scarce environment. The research for a new reliable linkage method that has less dependency on the amount of data may be an appropriate direction in the future.

Second, the DBN modeling tends to be subjective by determining the causal relationships of risk events with an empirical designation. In the DFBN models, the DBNs are used for establishing the temporal and systemic risks. However, the causal relationships of risk events are usually vague for human's cognition. Thus, identifying those relationships by practitioner's empirical judgment may not generate the genuine image of the real situation. According to Kjaerulff & Madsen (2008), data-driven BBNs can be applied to effectively reduce the influence of the subjectiveness of the model. The trade-off of a data-driven model is the high dependency on the quality, quantity and reliability of the data. The investigation of the data-driven DBNs is also a promising track of strengthening the DFBN models.

Third, a more user-friendly model execution interface may be developed for practitioner's application. At the current stage, an executable MATLAB program is available for each of the case studies, respectively. By modifying the MATLAB code, application cases with similar scenarios and input data structure can be realized. However, the expandability of the DFBN models by applying the existing programs is limited. In the Appendices I – III, the source MATLAB code for those three case studies mentioned in the previous section is provided. In the future, a more user-friendly program package can be developed in the future to make the model more efficient for researchers and practitioners. Ideally, a program with a Graphical User Interface (GUI) can be developed to expand the functionality of the DFBN models to the next level. In such case, the DFBN models can be easily built by entering minimum parameters by the users.

Appendix I –MATLAB Code for the DFBN Case Study

File Name: Data_CPT_AMT.m

```
T = 12; % Simulation Running Time

% Wholesaler DBN CPT Data

MSD = [0.15 0.85];
RCA = [0.1 0.9];
WEP = [0.95 0.6 0.3 0.05 0.05 0.4 0.7 0.95];
ROL = [0.8 0.3 0.2 0.7];
WIL = [0.7 0.1 0.3 0.9];
RUO = [0.2 0.6 0.05 0.3 0.8 0.4 0.95 0.7];
MSDF = [0.66 0.78 0.34 0.22];
RUOF = [0.11 0.33 0.11 0.33 0.89 0.67 0.89 0.67];

% Retailer DBN CPT Data

WSD = [0.3 0.7];
CCA = [0.5 0.5];
REP = [0.9 0.1 0.7 0.2 0.1 0.9 0.3 0.8];
COL = [0.95 0.1 0.05 0.9];
RIL = [0.8 0.2 0.2 0.8];
CUO = [0.2 0.9 0.05 0.4 0.8 0.1 0.95 0.6];
WSDF = [0.66 0.78 0.34 0.22];
CUOF = [0.11 0.33 0.11 0.33 0.89 0.67 0.89 0.67];

% Amount Data
RUOA = [10 30 50 30 20 60 20 20 10 40 30 10];
CUOA = [40 50 30 10 30 70 80 30 50 40 30 20];

% Optimization Data
w1 = 0.2;
w2 = 0.8;
RPP = 3;

S = [10 10 10 10 10 10 10 10 10 10 10 10];
C = [500 500 500 500 500 500 500 500 500 500 500 500];
SCC = [1 1 1 1 1 1 1 1 1 1 1 1];
RIC = [210 210 210 210 210 210 210 210 210 210 210 210];

% VaR Data Generator
n_C = 1000; mu_C = 20; sigma_C = 20;
lambdaStart_C = 14; deltaStart_C = 14; xiStart_C = 0.05;

n_R = 1000; mu_R = 90; sigma_R = 80;
lambdaStart_R = 62; deltaStart_R = 60; xiStart_R = 0.03;

% % CEUOR Inputs
% y0_C = 20;
```

```
%
% % REUOR Inputs
% y0_R = 10;
```

File Name: Retailer_Unfilled_Amt_data.m

```
WAD = [10 30 50 30 20 60 20 20 10 40 30 10];

RPR = [62 73 84 68 47 99 123 62 119 73 78 128];
```

File Name: RetailerDBN_CustUnfOrd.m

```
function    ProbCustUnfOrd = RetailerDBN_CustUnfOrd(WSD, WSDF,
CCA, REP, COL, RIL, CUO, CUOF)

%#codegen
ss = 6; % slice size
intra = zeros(ss);
intra(1,3) = 1;
intra(2,4) = 1;
intra(3,5) = 1;
intra(4, [3 6]) = 1;
intra(5,6) = 1;

inter = zeros(ss);
inter(1,1) = 1;
inter(6,4) = 1;

onodes = []; % observed
dnodes = 1:ss; % discrete
ns = 2*ones(1,ss); % binary nodes

TS = 12; % Number of Time Slice
% Temporal Customer Order Level Probability
ProbCustUnfOrd.VRiskHi = zeros(1,TS);



% BBN (In order to get the probability value at t = 1)
bnet1 = mk_bnet(intra, ns, 'discrete', dnodes, 'observed',
onodes);

bnet1.CPD{1} = tabular_CPD(bnet1, 1, WSD);
bnet1.CPD{2} = tabular_CPD(bnet1, 2, CCA);
bnet1.CPD{3} = tabular_CPD(bnet1, 3, REP);
bnet1.CPD{4} = tabular_CPD(bnet1, 4, COL);
bnet1.CPD{5} = tabular_CPD(bnet1, 5, RIL);
bnet1.CPD{6} = tabular_CPD(bnet1, 6, CUO);
```

```
evidence1 = cell(1,ss);

evidence1{1} = 1;

for i = 2:6
    evidence1{i} = [];
end
engine1 = jtree_inf_engine(bnet1);
[engine1, loglik] = enter_evidence(engine1, evidence1);
marg = marginal_nodes(engine1, 6);
% marg.T     % Inference Answer
ProbCustUnfOrd.VRiskHi(1) = marg.T(1);



% DBN (In order to get the probability value at t = 2:TS)
eclass1 = [1 2 3 4 5 6];
eclass2 = [7 2 3 8 5 6];
% eclass = [eclass1 eclass2];

bnet2 = mk_dbn(intra, inter, ns, 'discrete', dnodes, 'eclass1',
eclass1, 'eclass2', eclass2);

bnet2.CPD{1} = tabular_CPD(bnet2, 1, WSD);
bnet2.CPD{2} = tabular_CPD(bnet2, 2, CCA);
bnet2.CPD{3} = tabular_CPD(bnet2, 3, REP);
bnet2.CPD{4} = tabular_CPD(bnet2, 4, COL);
bnet2.CPD{5} = tabular_CPD(bnet2, 5, RIL);
bnet2.CPD{6} = tabular_CPD(bnet2, 6, CUO);
bnet2.CPD{7} = tabular_CPD(bnet2, 7, WSDF);
bnet2.CPD{8} = tabular_CPD(bnet2, 10, CUOF);

for T = 2:TS;
    ev = sample_dbn(bnet2, T);
    evidence2 = cell(ss, T);
    evidence2(onodes,:) = ev(onodes,:); % all cells besides
onodes are empty
    engine2 = smoother_engine(jtree_2TBN_inf_engine(bnet2));
    % engine2 = smoother_engine(hmm_2TBN_inf_engine(bnet));
    engine2 = enter_evidence(engine2, evidence2);
    m = marginal_nodes(engine2, 6, T);
    m.T;     % Inference Answer
    ProbCustUnfOrd.VRiskHi(T) = m.T(1);
end

onodes1 = 1;

for T = 2
    evidence3 = cell(ss,T);
    evidence3{onodes1,T} = [1];
%   engine2 = smoother_engine(jtree_2TBN_inf_engine(bnet2));
    engine2 = smoother_engine(hmm_2TBN_inf_engine(bnet2));
```

```
    engine2 = enter_evidence(engine2, evidence3);
    m = marginal_nodes(engine2, 6, T);
%   m.T;     % Inference Answer
    ProbCustUnfOrd.VRiskHi(T) = m.T(1);
end

for T = 6:7
    evidence4 = cell(ss,T);
    evidence4{onodes1,T} = [2];
%   engine2 = smoother_engine(jtree_2TBN_inf_engine(bnet2));
    engine2 = smoother_engine(hmm_2TBN_inf_engine(bnet2));
    engine2 = enter_evidence(engine2, evidence4);
    m = marginal_nodes(engine2, 6, T);
%   m.T;     % Inference Answer
    ProbCustUnfOrd.VRiskHi(T) = m.T(1);
end

onodes2 = 2;

for T = [3 11]
    evidence5 = cell(ss,T);
    evidence5{onodes2,T} = [2];
%   engine2 = smoother_engine(jtree_2TBN_inf_engine(bnet2));
    engine2 = smoother_engine(hmm_2TBN_inf_engine(bnet2));
    engine2 = enter_evidence(engine2, evidence5);
    m = marginal_nodes(engine2, 6, T);
%   m.T;     % Inference Answer
    ProbCustUnfOrd.VRiskHi(T) = m.T(1);
end

for T = [6 12]
    evidence6 = cell(ss,T);
    evidence6{onodes2,T} = [1];
%   engine2 = smoother_engine(jtree_2TBN_inf_engine(bnet2));
    engine2 = smoother_engine(hmm_2TBN_inf_engine(bnet2));
    engine2 = enter_evidence(engine2, evidence6);
    m = marginal_nodes(engine2, 6, T);
%   m.T;     % Inference Answer
    ProbCustUnfOrd.VRiskHi(T) = m.T(1);
end

% Illustration

% isbox = zeros(ss,1); isbox(dnodes) = 1;
% unfold = 4;
% draw_dbn(intra, inter, 0, unfold, {'a', 'b', 'c', 'd', 'e',
'f'}, isbox); % Draw DBN Diagram

ProbCustUnfOrd = ProbCustUnfOrd.VRiskHi;
```

File Name: RetailerDBN_CustOrdLvl.m

```
function    ProbCustOrdLvl = RetailerDBN_CustOrdLvl(WSD, WSDF,
CCA, REP, COL, RIL, CUO, CUOF)

%#codegen
ss = 6; % slice size
intra = zeros(ss);
intra(1,3) = 1;
intra(2,4) = 1;
intra(3,5) = 1;
intra(4, [3 6]) = 1;
intra(5,6) = 1;

inter = zeros(ss);
inter(1,1) = 1;
inter(6,4) = 1;

onodes = []; % observed
dnodes = 1:ss; % discrete
ns = 2*ones(1,ss); % binary nodes

TS = 12; % Number of Time Slice
% Temporal Customer Order Level Probability
ProbCustOrdLvl.VRiskHi = zeros(1,TS);



% BBN (In order to get the probability value at t = 1)
bnet1 = mk_bnet(intra, ns, 'discrete', dnodes, 'observed',
onodes);

bnet1.CPD{1} = tabular_CPD(bnet1, 1, WSD);
bnet1.CPD{2} = tabular_CPD(bnet1, 2, CCA);
bnet1.CPD{3} = tabular_CPD(bnet1, 3, REP);
bnet1.CPD{4} = tabular_CPD(bnet1, 4, COL);
bnet1.CPD{5} = tabular_CPD(bnet1, 5, RIL);
bnet1.CPD{6} = tabular_CPD(bnet1, 6, CUO);

evidence1 = cell(1,ss);

evidence1{1} = 1;

for i = 2:6
    evidence1{i} = [];
end
engine1 = jtree_inf_engine(bnet1);
[engine1, loglik] = enter_evidence(engine1, evidence1);
marg = marginal_nodes(engine1, 4);
% marg.T    % Inference Answer
ProbCustOrdLvl.VRiskHi(1) = marg.T(2);
```

```
% DBN (In order to get the probability value at t = 2:TS)
eclass1 = [1 2 3 4 5 6];
eclass2 = [7 2 3 8 5 6];
% eclass = [eclass1 eclass2];

bnet2 = mk_dbn(intra, inter, ns, 'discrete', dnodes, 'eclass1',
eclass1, 'eclass2', eclass2);

bnet2.CPD{1} = tabular_CPD(bnet2, 1, WSD);
bnet2.CPD{2} = tabular_CPD(bnet2, 2, CCA);
bnet2.CPD{3} = tabular_CPD(bnet2, 3, REP);
bnet2.CPD{4} = tabular_CPD(bnet2, 4, COL);
bnet2.CPD{5} = tabular_CPD(bnet2, 5, RIL);
bnet2.CPD{6} = tabular_CPD(bnet2, 6, CUO);
bnet2.CPD{7} = tabular_CPD(bnet2, 7, WSDF);
bnet2.CPD{8} = tabular_CPD(bnet2, 10, CUOF);

for T = 2:TS;
    ev = sample_dbn(bnet2, T);
    evidence2 = cell(ss, T);
    evidence2(onodes,:) = ev(onodes,:); % all cells besides
onodes are empty
    engine2 = smoother_engine(jtree_2TBN_inf_engine(bnet2));
    % engine2 = smoother_engine(hmm_2TBN_inf_engine(bnet));
    engine2 = enter_evidence(engine2, evidence2);
    m = marginal_nodes(engine2, 4, T);
    m.T;    % Inference Answer
    ProbCustOrdLvl.VRiskHi(T) = m.T(2);
end

onodes1 = 1;

for T = 2
    evidence3 = cell(ss,T);
    evidence3{onodes1,T} = [1];
%   engine2 = smoother_engine(jtree_2TBN_inf_engine(bnet2));
    engine2 = smoother_engine(hmm_2TBN_inf_engine(bnet2));
    engine2 = enter_evidence(engine2, evidence3);
    m = marginal_nodes(engine2, 4, T);
%   m.T;    % Inference Answer
    ProbCustOrdLvl.VRiskHi(T) = m.T(2);
end

for T = 6:7
    evidence4 = cell(ss,T);
    evidence4{onodes1,T} = [2];
%   engine2 = smoother_engine(jtree_2TBN_inf_engine(bnet2));
    engine2 = smoother_engine(hmm_2TBN_inf_engine(bnet2));
    engine2 = enter_evidence(engine2, evidence4);
    m = marginal_nodes(engine2, 4, T);
```

```
%   m.T;    % Inference Answer
    ProbCustOrdLvl.VRiskHi(T) = m.T(2);
end

onodes2 = 2;

for T = [3 11]
    evidence5 = cell(ss,T);
    evidence5{onodes2,T} = [2];
%   engine2 = smoother_engine(jtree_2TBN_inf_engine(bnet2));
    engine2 = smoother_engine(hmm_2TBN_inf_engine(bnet2));
    engine2 = enter_evidence(engine2, evidence5);
    m = marginal_nodes(engine2, 4, T);
%   m.T;    % Inference Answer
    ProbCustOrdLvl.VRiskHi(T) = m.T(2);
end

for T = [6 12]
    evidence6 = cell(ss,T);
    evidence6{onodes2,T} = [1];
%   engine2 = smoother_engine(jtree_2TBN_inf_engine(bnet2));
    engine2 = smoother_engine(hmm_2TBN_inf_engine(bnet2));
    engine2 = enter_evidence(engine2, evidence6);
    m = marginal_nodes(engine2, 4, T);
%   m.T;    % Inference Answer
    ProbCustOrdLvl.VRiskHi(T) = m.T(2);
end
% Illustration

% isbox = zeros(ss,1); isbox(dnodes) = 1;
% unfold = 4;
% draw_dbn(intra, inter, 0, unfold, {'a', 'b', 'c', 'd', 'e',
'f'}, isbox); % Draw DBN Diagram

ProbCustOrdLvl = ProbCustOrdLvl.VRiskHi;
```

File Name: WholesalerDBN.m

```
function  ProbRetUnfOrd = WholesalerDBN(MSD, MSDF, RCA, WEP, ROL,
WIL, RUO, RUOF)

%#codegen
ss = 6; % slice size
intra = zeros(ss);
intra(1,3) = 1;
intra(2,4) = 1;
intra(3,5) = 1;
intra(4, [3 6]) = 1;
intra(5,6) = 1;
```

```
inter = zeros(ss);
inter(1,1) = 1;
inter(6,4) = 1;

onodes = []; % observed
dnodes = 1:ss; % discrete
ns = 2*ones(1,ss); % binary nodes

TS = 12; % Number of Time Slice
% Temporal Customer Order Level Probability
ProbRetUnfOrd.VRiskHi = zeros(1,TS);



% BBN (In order to get the probability value at t = 1)
bnet1 = mk_bnet(intra, ns, 'discrete', dnodes, 'observed',
onodes);

bnet1.CPD{1} = tabular_CPD(bnet1, 1, MSD);
bnet1.CPD{2} = tabular_CPD(bnet1, 2, RCA);
bnet1.CPD{3} = tabular_CPD(bnet1, 3, WEP);
bnet1.CPD{4} = tabular_CPD(bnet1, 4, ROL);
bnet1.CPD{5} = tabular_CPD(bnet1, 5, WIL);
bnet1.CPD{6} = tabular_CPD(bnet1, 6, RUO);

evidence1 = cell(1,ss);

for i = 1
    evidence1{i} = 1;
end

for i = 2
    evidence1{i} = 2;
end

for i = 3:6
    evidence1{i} = [];
end
engine1 = jtree_inf_engine(bnet1);
[engine1, loglik] = enter_evidence(engine1, evidence1);
marg = marginal_nodes(engine1, 6);

ProbRetUnfOrd.VRiskHi(1) = marg.T(1);



% DBN (In order to get the probability value at t = 2:TS)
eclass1 = [1 2 3 4 5 6];
eclass2 = [7 2 3 8 5 6];
% eclass = [eclass1 eclass2];
```

```
bnet2 = mk_dbn(intra, inter, ns, 'discrete', dnodes, 'eclass1',
eclass1, 'eclass2', eclass2);

bnet2.CPD{1} = tabular_CPD(bnet2, 1, MSD);
bnet2.CPD{2} = tabular_CPD(bnet2, 2, RCA);
bnet2.CPD{3} = tabular_CPD(bnet2, 3, WEP);
bnet2.CPD{4} = tabular_CPD(bnet2, 4, ROL);
bnet2.CPD{5} = tabular_CPD(bnet2, 5, WIL);
bnet2.CPD{6} = tabular_CPD(bnet2, 6, RUO);
bnet2.CPD{7} = tabular_CPD(bnet2, 7, MSDF);
bnet2.CPD{8} = tabular_CPD(bnet2, 10, RUOF);

for T = 2:TS
    ev = sample_dbn(bnet2, T);
    evidence2 = cell(ss,T);
    evidence2(onodes,:) = ev(onodes, :); % all cells besides
onodes are empty
    engine2 = smoother_engine(jtree_2TBN_inf_engine(bnet2));
    % engine2 = smoother_engine(hmm_2TBN_inf_engine(bnet2));
    engine2 = enter_evidence(engine2, evidence2);
    m = marginal_nodes(engine2, 6, T);
    m.T;     % Inference Answer
    ProbRetUnfOrd.VRiskHi(T) = m.T(1);
end

onodes1 = 1;

for T = 2
    evidence3 = cell(ss,T);
    evidence3{onodes1,T} = [1];
%   engine2 = smoother_engine(jtree_2TBN_inf_engine(bnet2));
    engine2 = smoother_engine(hmm_2TBN_inf_engine(bnet2));
    engine2 = enter_evidence(engine2, evidence3);
    m = marginal_nodes(engine2, 6, T);
%   m.T;     % Inference Answer
    ProbRetUnfOrd.VRiskHi(T) = m.T(1);
end

for T = [5 9 10]
    evidence4 = cell(ss,T);
    evidence4{onodes1,T} = [2];
%   engine2 = smoother_engine(jtree_2TBN_inf_engine(bnet2));
    engine2 = smoother_engine(hmm_2TBN_inf_engine(bnet2));
    engine2 = enter_evidence(engine2, evidence4);
    m = marginal_nodes(engine2, 6, T);
%   m.T;     % Inference Answer
    ProbRetUnfOrd.VRiskHi(T) = m.T(1);
end

onodes2 = 2;

for T = [9 12]
```

```
    evidence5 = cell(ss,T);
    evidence5{onodes2,T} = [2];
%   engine2 = smoother_engine(jtree_2TBN_inf_engine(bnet2));
    engine2 = smoother_engine(hmm_2TBN_inf_engine(bnet2));
    engine2 = enter_evidence(engine2, evidence5);
    m = marginal_nodes(engine2, 6, T);
%   m.T;     % Inference Answer
    ProbRetUnfOrd.VRiskHi(T) = m.T(1);
end

for T = [8 11]
    evidence6 = cell(ss,T);
    evidence6{onodes2,T} = [1];
%   engine2 = smoother_engine(jtree_2TBN_inf_engine(bnet2));
    engine2 = smoother_engine(hmm_2TBN_inf_engine(bnet2));
    engine2 = enter_evidence(engine2, evidence6);
    m = marginal_nodes(engine2, 6, T);
%   m.T;     % Inference Answer
    ProbRetUnfOrd.VRiskHi(T) = m.T(1);
end

% Illustration

% isbox = zeros(ss,1); isbox(dnodes) = 1;
% unfold = 4;
% draw_dbn(intra, inter, 0, unfold, {'a', 'b', 'c', 'd', 'e',
'f'}, isbox); % Draw DBN Diagram

ProbRetUnfOrd = ProbRetUnfOrd.VRiskHi;
```

File Name: Retailer_Exp_Unf_Rate2.m

```
% Wholesaler Part (Output: Retailer Unfilled Orders Risk High)
WRiskHi = WholesalerDBN(MSD, MSDF, RCA, WEP, ROL, WIL, RUO,
RUOF);

% Retailer Part (Output: Customer Unfilled Orders Risk High,
                           % Customer Order Level Probability High)
RCOLRiskHi = RetailerDBN_CustOrdLvl(WSD, WSDF, CCA, REP, COL,
RIL, CUO, CUOF);
RCUORiskHi = RetailerDBN_CustUnfOrd(WSD, WSDF, CCA, REP, COL,
RIL, CUO, CUOF);

% Retailer's Expected Unfilled Order Rate =
% Probability of Retailer's Unfilled Order * Retailer's Unfilled
Order Amount
REUOR = WRiskHi.*RUOA;

WI = zeros(1,12);                % Wholesaler's Inventory Level
```

```
WPR = zeros(1,12);              % Wholesaler's Procurement Rate

% Customer's Expected Unfilled Order Rate =
% Probability of Customer's Unfilled Order * Customer's Unfilled
Order Amount
CEUOR = RCUORiskHi.*CUOA;

RI = zeros(1,12);               % Retailer's Inventory Level
CPR = zeros(1,12);              % Customer's Purchase Rate
RPR = zeros(1,12);              % Retailer's Procurement Rate
WSR = zeros(1,12);
RSR = zeros(1,12);

WI(1) = 2000;                   % Initial Value = 2000 Units
RI(1) = 200;                    % Initial Value = 200 Units

% Initial Values for Other Variables
CPR(1) = RCOLRiskHi(1)*100 + (1 - RCOLRiskHi(1))*50;
RSR(1) = CPR(1) - CEUOR(1);
RPR(1) = RSR(1) + 15;
WSR(1) = RPR(1) - REUOR(1);
WPR(1) = WSR(1);

for i = 2:12

% Wholesaler's Inventory = Wholesaler Procurement Rate - Retailer
Procurement Rate
WI(i) = WI(i-1) + WPR(i-1) - RPR(i-1);

% Retailer's Inventory = Retailer Procurement Rate - Customer
Purchase Rate
RI(i) = RI(i-1) + RPR(i-1) - CPR(i-1);

% Customer's Purchase Rate =
% Probability of Customer Order Level High*100+(1-Probability of
Customer Order Level High)*50
CPR(i) = RCOLRiskHi(i)*100 + (1 - RCOLRiskHi(i))*50;

% Retailer Sales Rate = Customer Purchase Rate - Customer's
Expected Unfilled Order Rate
RSR(i) = CPR(i) - CEUOR(i);

% Retailer Procurement Rate Plan
if RI(i) <= 150;
    RPR(i) = RSR(i) + 50;
else RPR(i) = RSR(i);% + 15;
end

% Wholesaler Sales Rate = Retailer Procurement Rate -
% Retailer's Expected Unfilled Order Rate
WSR(i) = RPR(i) - REUOR(i);
```

```
% Wholesaler Procument Rate Plan
if WI(i) <= 1950;
    WPR(i) = WSR(i) + 100;
else WPR(i) = WSR(i);
end

end

WI

RI


% Figures
x = 1:12;
figure
plot(x, WI, '--or')
axis([0 12 0 2500])

figure
plot(x, RI, '-*k')
axis([0 12 0 250])
```

Appendix II –MATLAB Code for the ODFBN Case Study

File Name: Data_CPT_AMT.m

```
T = 12; % Simulation Running Time

% Wholesaler DBN CPT Data

MSD = [0.15 0.85];
RCA = [0.1 0.9];
WEP = [0.95 0.6 0.3 0.05 0.05 0.4 0.7 0.95];
ROL = [0.8 0.3 0.2 0.7];
WIL = [0.7 0.1 0.3 0.9];
RUO = [0.2 0.6 0.05 0.3 0.8 0.4 0.95 0.7];
MSDF = [0.66 0.78 0.34 0.22];
RUOF = [0.11 0.33 0.11 0.33 0.89 0.67 0.89 0.67];

% Retailer DBN CPT Data

WSD = [0.3 0.7];
CCA = [0.5 0.5];
REP = [0.9 0.1 0.7 0.2 0.1 0.9 0.3 0.8];
COL = [0.95 0.1 0.05 0.9];
RIL = [0.8 0.2 0.2 0.8];
CUO = [0.2 0.9 0.05 0.4 0.8 0.1 0.95 0.6];
WSDF = [0.66 0.78 0.34 0.22];
CUOF = [0.11 0.33 0.11 0.33 0.89 0.67 0.89 0.67];

% Amount Data
RUOA = [10 30 50 30 20 60 20 20 10 40 30 10];
CUOA = [40 50 30 10 30 70 80 30 50 40 30 20];

% Optimization Data
w1 = 0.2;
w2 = 0.8;
RPP = 3;

S = [10 10 10 10 10 10 10 10 10 10 10 10];
C = [500 500 500 500 500 500 500 500 500 500 500 500];
SCC = [1 1 1 1 1 1 1 1 1 1 1 1];
RIC = [210 210 210 210 210 210 210 210 210 210 210 210];

% VaR Data Generator
n_C = 1000; mu_C = 20; sigma_C = 20;
lambdaStart_C = 14; deltaStart_C = 14; xiStart_C = 0.05;

n_R = 1000; mu_R = 90; sigma_R = 80;
lambdaStart_R = 62; deltaStart_R = 60; xiStart_R = 0.03;

% % CEUOR Inputs
% y0_C = 20;
%
```

```
% % REUOR Inputs
% y0_R = 10;
```

File Name: VaR_REUOR.m

```
function pdf_gevd_new = VaR_REUOR(y_R,y0_R)

WRiskHi = WholesalerDBN(MSD, MSDF, RCA, WEP, ROL, WIL, RUO,
RUOF);

pdf_gevd_new = exp(-(1-xiHat*((y_R-
lambdaHat)/deltaHat)).^(1/xiHat)) - WRiskHi;

fun = @Var_REUOR;

REUOR = fzero(fun,y0_R);

pdf_gevd_new = REUOR;
```

File Name: VaR_CEUOR.m

```
function pdf_gevd_new = VaR_CEUOR(y_C,y0_C)

RCUORiskHi = RetailerDBN_CustUnfOrd(WSD, WSDF, CCA, REP, COL,
RIL, CUO, CUOF);

pdf_gevd_new = exp(-(1-xiHat*((y_C-
lambdaHat)/deltaHat)).^(1/xiHat)) - RCUORiskHi;

fun = @Var_REUOR;

CEUOR = fzero(fun,y0_C);

pdf_gevd_new = CEUOR;
```

File Name: RetailerDBN_CustOrdLvl.m

```
function    ProbCustOrdLvl = RetailerDBN_CustOrdLvl(WSD, WSDF,
CCA, REP, COL, RIL, CUO, CUOF)

%#codegen
ss = 6; % slice size
intra = zeros(ss);
```

```
intra(1,3) = 1;
intra(2,4) = 1;
intra(3,5) = 1;
intra(4, [3 6]) = 1;
intra(5,6) = 1;

inter = zeros(ss);
inter(1,1) = 1;
inter(6,4) = 1;

onodes = []; % observed
dnodes = 1:ss; % discrete
ns = 2*ones(1,ss); % binary nodes

TS = 12; % Number of Time Slice
% Temporal Customer Order Level Probability
ProbCustOrdLvl.VRiskHi = zeros(1,TS);



% BBN (In order to get the probability value at t = 1)
bnet1 = mk_bnet(intra, ns, 'discrete', dnodes, 'observed',
onodes);

bnet1.CPD{1} = tabular_CPD(bnet1, 1, WSD);
bnet1.CPD{2} = tabular_CPD(bnet1, 2, CCA);
bnet1.CPD{3} = tabular_CPD(bnet1, 3, REP);
bnet1.CPD{4} = tabular_CPD(bnet1, 4, COL);
bnet1.CPD{5} = tabular_CPD(bnet1, 5, RIL);
bnet1.CPD{6} = tabular_CPD(bnet1, 6, CUO);

evidence1 = cell(1,ss);

evidence1{1} = 1;

for i = 2:6
    evidence1{i} = [];
end
engine1 = jtree_inf_engine(bnet1);
[engine1, loglik] = enter_evidence(engine1, evidence1);
marg = marginal_nodes(engine1, 4);
% marg.T    % Inference Answer
ProbCustOrdLvl.VRiskHi(1) = marg.T(2);



% DBN (In order to get the probability value at t = 2:TS)
eclass1 = [1 2 3 4 5 6];
eclass2 = [7 2 3 8 5 6];
% eclass = [eclass1 eclass2];
```

```
bnet2 = mk_dbn(intra, inter, ns, 'discrete', dnodes, 'eclass1',
eclass1, 'eclass2', eclass2);

bnet2.CPD{1} = tabular_CPD(bnet2, 1, WSD);
bnet2.CPD{2} = tabular_CPD(bnet2, 2, CCA);
bnet2.CPD{3} = tabular_CPD(bnet2, 3, REP);
bnet2.CPD{4} = tabular_CPD(bnet2, 4, COL);
bnet2.CPD{5} = tabular_CPD(bnet2, 5, RIL);
bnet2.CPD{6} = tabular_CPD(bnet2, 6, CUO);
bnet2.CPD{7} = tabular_CPD(bnet2, 7, WSDF);
bnet2.CPD{8} = tabular_CPD(bnet2, 10, CUOF);

for T = 2:TS;
    ev = sample_dbn(bnet2, T);
    evidence2 = cell(ss, T);
    evidence2(onodes,:) = ev(onodes,:); % all cells besides
onodes are empty
    engine2 = smoother_engine(jtree_2TBN_inf_engine(bnet2));
    % engine2 = smoother_engine(hmm_2TBN_inf_engine(bnet));
    engine2 = enter_evidence(engine2, evidence2);
    m = marginal_nodes(engine2, 4, T);
    m.T;    % Inference Answer
    ProbCustOrdLvl.VRiskHi(T) = m.T(2);
end

onodes1 = 1;

for T = 2
    evidence3 = cell(ss,T);
    evidence3{onodes1,T} = [1];
%   engine2 = smoother_engine(jtree_2TBN_inf_engine(bnet2));
    engine2 = smoother_engine(hmm_2TBN_inf_engine(bnet2));
    engine2 = enter_evidence(engine2, evidence3);
    m = marginal_nodes(engine2, 4, T);
%   m.T;    % Inference Answer
    ProbCustOrdLvl.VRiskHi(T) = m.T(2);
end

for T = 6:7
    evidence4 = cell(ss,T);
    evidence4{onodes1,T} = [2];
%   engine2 = smoother_engine(jtree_2TBN_inf_engine(bnet2));
    engine2 = smoother_engine(hmm_2TBN_inf_engine(bnet2));
    engine2 = enter_evidence(engine2, evidence4);
    m = marginal_nodes(engine2, 4, T);
%   m.T;    % Inference Answer
    ProbCustOrdLvl.VRiskHi(T) = m.T(2);
end

onodes2 = 2;

for T = [3 11]
```

```
    evidence5 = cell(ss,T);
    evidence5{onodes2,T} = [2];
%   engine2 = smoother_engine(jtree_2TBN_inf_engine(bnet2));
    engine2 = smoother_engine(hmm_2TBN_inf_engine(bnet2));
    engine2 = enter_evidence(engine2, evidence5);
    m = marginal_nodes(engine2, 4, T);
%   m.T;     % Inference Answer
    ProbCustOrdLvl.VRiskHi(T) = m.T(2);
end

for T = [6 12]
    evidence6 = cell(ss,T);
    evidence6{onodes2,T} = [1];
%   engine2 = smoother_engine(jtree_2TBN_inf_engine(bnet2));
    engine2 = smoother_engine(hmm_2TBN_inf_engine(bnet2));
    engine2 = enter_evidence(engine2, evidence6);
    m = marginal_nodes(engine2, 4, T);
%   m.T;     % Inference Answer
    ProbCustOrdLvl.VRiskHi(T) = m.T(2);
end
% Illustration

% isbox = zeros(ss,1); isbox(dnodes) = 1;
% unfold = 4;
% draw_dbn(intra, inter, 0, unfold, {'a', 'b', 'c', 'd', 'e',
'f'}, isbox); % Draw DBN Diagram

ProbCustOrdLvl = ProbCustOrdLvl.VRiskHi;
```

File Name: RetailerDBN_CustUnfOrd.m

```
function    ProbCustUnfOrd = RetailerDBN_CustUnfOrd(WSD, WSDF,
CCA, REP, COL, RIL, CUO, CUOF)

%#codegen
ss = 6; % slice size
intra = zeros(ss);
intra(1,3) = 1;
intra(2,4) = 1;
intra(3,5) = 1;
intra(4, [3 6]) = 1;
intra(5,6) = 1;

inter = zeros(ss);
inter(1,1) = 1;
inter(6,4) = 1;

onodes = []; % observed
dnodes = 1:ss; % discrete
```

```
ns = 2*ones(1,ss); % binary nodes

TS = 12; % Number of Time Slice
% Temporal Customer Order Level Probability
ProbCustUnfOrd.VRiskHi = zeros(1,TS);



% BBN (In order to get the probability value at t = 1)
bnet1 = mk_bnet(intra, ns, 'discrete', dnodes, 'observed',
onodes);

bnet1.CPD{1} = tabular_CPD(bnet1, 1, WSD);
bnet1.CPD{2} = tabular_CPD(bnet1, 2, CCA);
bnet1.CPD{3} = tabular_CPD(bnet1, 3, REP);
bnet1.CPD{4} = tabular_CPD(bnet1, 4, COL);
bnet1.CPD{5} = tabular_CPD(bnet1, 5, RIL);
bnet1.CPD{6} = tabular_CPD(bnet1, 6, CUO);

evidence1 = cell(1,ss);

evidence1{1} = 1;

for i = 2:6
    evidence1{i} = [];
end
engine1 = jtree_inf_engine(bnet1);
[engine1, loglik] = enter_evidence(engine1, evidence1);
marg = marginal_nodes(engine1, 6);
% marg.T    % Inference Answer
ProbCustUnfOrd.VRiskHi(1) = marg.T(1);



% DBN (In order to get the probability value at t = 2:TS)
eclass1 = [1 2 3 4 5 6];
eclass2 = [7 2 3 8 5 6];
% eclass = [eclass1 eclass2];

bnet2 = mk_dbn(intra, inter, ns, 'discrete', dnodes, 'eclass1',
eclass1, 'eclass2', eclass2);

bnet2.CPD{1} = tabular_CPD(bnet2, 1, WSD);
bnet2.CPD{2} = tabular_CPD(bnet2, 2, CCA);
bnet2.CPD{3} = tabular_CPD(bnet2, 3, REP);
bnet2.CPD{4} = tabular_CPD(bnet2, 4, COL);
bnet2.CPD{5} = tabular_CPD(bnet2, 5, RIL);
bnet2.CPD{6} = tabular_CPD(bnet2, 6, CUO);
bnet2.CPD{7} = tabular_CPD(bnet2, 7, WSDF);
bnet2.CPD{8} = tabular_CPD(bnet2, 10, CUOF);

for T = 2:TS;
```

```
    ev = sample_dbn(bnet2, T);
    evidence2 = cell(ss, T);
    evidence2(onodes,:) = ev(onodes,:); % all cells besides
onodes are empty
    engine2 = smoother_engine(jtree_2TBN_inf_engine(bnet2));
    % engine2 = smoother_engine(hmm_2TBN_inf_engine(bnet));
    engine2 = enter_evidence(engine2, evidence2);
    m = marginal_nodes(engine2, 6, T);
    m.T;     % Inference Answer
    ProbCustUnfOrd.VRiskHi(T) = m.T(1);
end

onodes1 = 1;

for T = 2
    evidence3 = cell(ss,T);
    evidence3{onodes1,T} = [1];
%   engine2 = smoother_engine(jtree_2TBN_inf_engine(bnet2));
    engine2 = smoother_engine(hmm_2TBN_inf_engine(bnet2));
    engine2 = enter_evidence(engine2, evidence3);
    m = marginal_nodes(engine2, 6, T);
%   m.T;     % Inference Answer
    ProbCustUnfOrd.VRiskHi(T) = m.T(1);
end

for T = 6:7
    evidence4 = cell(ss,T);
    evidence4{onodes1,T} = [2];
%   engine2 = smoother_engine(jtree_2TBN_inf_engine(bnet2));
    engine2 = smoother_engine(hmm_2TBN_inf_engine(bnet2));
    engine2 = enter_evidence(engine2, evidence4);
    m = marginal_nodes(engine2, 6, T);
%   m.T;     % Inference Answer
    ProbCustUnfOrd.VRiskHi(T) = m.T(1);
end

onodes2 = 2;

for T = [3 11]
    evidence5 = cell(ss,T);
    evidence5{onodes2,T} = [2];
%   engine2 = smoother_engine(jtree_2TBN_inf_engine(bnet2));
    engine2 = smoother_engine(hmm_2TBN_inf_engine(bnet2));
    engine2 = enter_evidence(engine2, evidence5);
    m = marginal_nodes(engine2, 6, T);
%   m.T;     % Inference Answer
    ProbCustUnfOrd.VRiskHi(T) = m.T(1);
end

for T = [6 12]
    evidence6 = cell(ss,T);
    evidence6{onodes2,T} = [1];
```

```
%   engine2 = smoother_engine(jtree_2TBN_inf_engine(bnet2));
    engine2 = smoother_engine(hmm_2TBN_inf_engine(bnet2));
    engine2 = enter_evidence(engine2, evidence6);
    m = marginal_nodes(engine2, 6, T);
%   m.T;    % Inference Answer
    ProbCustUnfOrd.VRiskHi(T) = m.T(1);
end

% Illustration

% isbox = zeros(ss,1); isbox(dnodes) = 1;
% unfold = 4;
% draw_dbn(intra, inter, 0, unfold, {'a', 'b', 'c', 'd', 'e',
'f'}, isbox); % Draw DBN Diagram

ProbCustUnfOrd = ProbCustUnfOrd.VRiskHi;
```

File Name: WholesalerDBN.m

```
function  ProbRetUnfOrd = WholesalerDBN(MSD, MSDF, RCA, WEP, ROL,
WIL, RUO, RUOF)

%#codegen
ss = 6; % slice size
intra = zeros(ss);
intra(1,3) = 1;
intra(2,4) = 1;
intra(3,5) = 1;
intra(4, [3 6]) = 1;
intra(5,6) = 1;

inter = zeros(ss);
inter(1,1) = 1;
inter(6,4) = 1;

onodes = []; % observed
dnodes = 1:ss; % discrete
ns = 2*ones(1,ss); % binary nodes

TS = 12; % Number of Time Slice
% Temporal Customer Order Level Probability
ProbRetUnfOrd.VRiskHi = zeros(1,TS);



% BBN (In order to get the probability value at t = 1)
bnet1 = mk_bnet(intra, ns, 'discrete', dnodes, 'observed',
onodes);
```

```
bnet1.CPD{1} = tabular_CPD(bnet1, 1, MSD);
bnet1.CPD{2} = tabular_CPD(bnet1, 2, RCA);
bnet1.CPD{3} = tabular_CPD(bnet1, 3, WEP);
bnet1.CPD{4} = tabular_CPD(bnet1, 4, ROL);
bnet1.CPD{5} = tabular_CPD(bnet1, 5, WIL);
bnet1.CPD{6} = tabular_CPD(bnet1, 6, RUO);

evidence1 = cell(1,ss);

for i = 1
    evidence1{i} = 1;
end

for i = 2
    evidence1{i} = 2;
end

for i = 3:6
    evidence1{i} = [];
end
engine1 = jtree_inf_engine(bnet1);
[engine1, loglik] = enter_evidence(engine1, evidence1);
marg = marginal_nodes(engine1, 6);

ProbRetUnfOrd.VRiskHi(1) = marg.T(1);



% DBN (In order to get the probability value at t = 2:TS)
eclass1 = [1 2 3 4 5 6];
eclass2 = [7 2 3 8 5 6];
% eclass = [eclass1 eclass2];

bnet2 = mk_dbn(intra, inter, ns, 'discrete', dnodes, 'eclass1',
eclass1, 'eclass2', eclass2);

bnet2.CPD{1} = tabular_CPD(bnet2, 1, MSD);
bnet2.CPD{2} = tabular_CPD(bnet2, 2, RCA);
bnet2.CPD{3} = tabular_CPD(bnet2, 3, WEP);
bnet2.CPD{4} = tabular_CPD(bnet2, 4, ROL);
bnet2.CPD{5} = tabular_CPD(bnet2, 5, WIL);
bnet2.CPD{6} = tabular_CPD(bnet2, 6, RUO);
bnet2.CPD{7} = tabular_CPD(bnet2, 7, MSDF);
bnet2.CPD{8} = tabular_CPD(bnet2, 10, RUOF);

for T = 2:TS
    ev = sample_dbn(bnet2, T);
    evidence2 = cell(ss,T);
    evidence2(onodes,:) = ev(onodes, :); % all cells besides
onodes are empty
    engine2 = smoother_engine(jtree_2TBN_inf_engine(bnet2));
    % engine2 = smoother_engine(hmm_2TBN_inf_engine(bnet2));
```

```
    engine2 = enter_evidence(engine2, evidence2);
    m = marginal_nodes(engine2, 6, T);
    m.T;     % Inference Answer
    ProbRetUnfOrd.VRiskHi(T) = m.T(1);
end

onodes1 = 1;

for T = 2
    evidence3 = cell(ss,T);
    evidence3{onodes1,T} = [1];
%   engine2 = smoother_engine(jtree_2TBN_inf_engine(bnet2));
    engine2 = smoother_engine(hmm_2TBN_inf_engine(bnet2));
    engine2 = enter_evidence(engine2, evidence3);
    m = marginal_nodes(engine2, 6, T);
%   m.T;     % Inference Answer
    ProbRetUnfOrd.VRiskHi(T) = m.T(1);
end

for T = [5 9 10]
    evidence4 = cell(ss,T);
    evidence4{onodes1,T} = [2];
%   engine2 = smoother_engine(jtree_2TBN_inf_engine(bnet2));
    engine2 = smoother_engine(hmm_2TBN_inf_engine(bnet2));
    engine2 = enter_evidence(engine2, evidence4);
    m = marginal_nodes(engine2, 6, T);
%   m.T;     % Inference Answer
    ProbRetUnfOrd.VRiskHi(T) = m.T(1);
end

onodes2 = 2;

for T = [9 12]
    evidence5 = cell(ss,T);
    evidence5{onodes2,T} = [2];
%   engine2 = smoother_engine(jtree_2TBN_inf_engine(bnet2));
    engine2 = smoother_engine(hmm_2TBN_inf_engine(bnet2));
    engine2 = enter_evidence(engine2, evidence5);
    m = marginal_nodes(engine2, 6, T);
%   m.T;     % Inference Answer
    ProbRetUnfOrd.VRiskHi(T) = m.T(1);
end

for T = [8 11]
    evidence6 = cell(ss,T);
    evidence6{onodes2,T} = [1];
%   engine2 = smoother_engine(jtree_2TBN_inf_engine(bnet2));
    engine2 = smoother_engine(hmm_2TBN_inf_engine(bnet2));
    engine2 = enter_evidence(engine2, evidence6);
    m = marginal_nodes(engine2, 6, T);
%   m.T;     % Inference Answer
    ProbRetUnfOrd.VRiskHi(T) = m.T(1);
```

```
end

% Illustration

% isbox = zeros(ss,1); isbox(dnodes) = 1;
% unfold = 4;
% draw_dbn(intra, inter, 0, unfold, {'a', 'b', 'c', 'd', 'e',
'f'}, isbox); % Draw DBN Diagram

ProbRetUnfOrd = ProbRetUnfOrd.VRiskHi;
```

File Name: mle_gevd_REUOR.m

```
% Generalized extreme value distributions (GEVD) Parameter
Estimate for
% REUOR
function REUOR =
mle_gevd_REUOR(T,n_R,mu_R,sigma_R,lambdaStart_R,deltaStart_R,xiSt
art_R,...
    MSD, MSDF, RCA, WEP, ROL, WIL, RUO, RUOF)

x_gen_R = normrnd(mu_R,sigma_R,n_R,1);

x_gen_R2 = zeros(n_R,1);

for i = 1:n_R

if x_gen_R(i) >= 0;
    x_gen_R2(i) = x_gen_R(i);
else x_gen_R2(i) = 0;
end

end

pdf_gevd_R = @(x,lambda,delta,xi) ...
    (1./delta).*(exp(-(1-xi.*((x-lambda)./delta)).^(1./xi))).*(1-
xi.*((x-lambda)./delta)).^((1./xi)-1);

start = [lambdaStart_R deltaStart_R xiStart_R];

lb = [];
ub = [];
gevd_paramEsts_R = mle(x_gen_R2, 'pdf',pdf_gevd_R,
'start',start, ...
    'lower',lb, 'upper',ub);

lambdaHat_R = gevd_paramEsts_R(1);
deltaHat_R = gevd_paramEsts_R(2);
xiHat_R = gevd_paramEsts_R(3);
```

```
% lambdaHat_C = round((gevd_paramEsts(1)*10000))/10000;
% deltaHat_C = round((gevd_paramEsts(1)*10000))/10000;
% xiHat_C = round((gevd_paramEsts(1)*10000))/10000;

WRiskHi = WholesalerDBN(MSD, MSDF, RCA, WEP, ROL, WIL, RUO,
RUOF);

REUOR = zeros(1,T);

syms x

cdf_gevd_R(x) = exp(-(1-xiHat_R.*((x-
lambdaHat_R)./deltaHat_R)).^(1./xiHat_R));

% inv_gevd_C = matlabFunction(finverse(cdf_gevd));

for i = 1:T

REUOR(i) = solve(cdf_gevd_R == WRiskHi(i),x);

end


% fzero method
% for i = 1:T
%
% fun = @(y_C)(exp(-((1-xiHat_C*((y_C-
lambdaHat_C)/deltaHat_C)).^(1/xiHat_C))) - RCUORiskHi(i));
%
% CEUOR(i) = fzero(fun,y0_C)
%
% end
```

File Name: mle_gevd_CEUOR.m

```
% Generalized extreme value distributions (GEVD) Parameter
Estimate for
% CEUOR
function CEUOR =
mle_gevd_CEUOR(T,n_C,mu_C,sigma_C,lambdaStart_C,deltaStart_C,xiSt
art_C,...
    WSD, WSDF, CCA, REP, COL, RIL, CUO, CUOF)

x_gen_C = normrnd(mu_C,sigma_C,n_C,1);

x_gen_C2 = zeros(n_C,1);

for i = 1:n_C
```

```
if x_gen_C(i) >= 0;
    x_gen_C2(i) = x_gen_C(i);
else x_gen_C2(i) = 0;
end

end

pdf_gevd_C = @(x,lambda,delta,xi) ...
    (1./delta).*(exp(-(1-xi.*((x-lambda)./delta)).^(1./xi))).*(1-
xi.*((x-lambda)./delta)).^((1./xi)-1);

start = [lambdaStart_C deltaStart_C xiStart_C];

lb = [];
ub = [];
gevd_paramEsts_C = mle(x_gen_C2, 'pdf',pdf_gevd_C,
'start',start, ...
    'lower',lb, 'upper',ub);

lambdaHat_C = gevd_paramEsts_C(1);
deltaHat_C = gevd_paramEsts_C(2);
xiHat_C = gevd_paramEsts_C(3);

% lambdaHat_C = round((gevd_paramEsts(1)*10000))/10000;
% deltaHat_C = round((gevd_paramEsts(1)*10000))/10000;
% xiHat_C = round((gevd_paramEsts(1)*10000))/10000;

RCUORiskHi = RetailerDBN_CustUnfOrd(WSD, WSDF, CCA, REP, COL,
RIL, CUO, CUOF);

CEUOR = zeros(1,T);

syms x

cdf_gevd_C(x) = exp(-(1-xiHat_C.*((x-
lambdaHat_C)./deltaHat_C)).^(1./xiHat_C));

% inv_gevd_C = matlabFunction(finverse(cdf_gevd));

for i = 1:T

CEUOR(i) = solve(cdf_gevd_C == RCUORiskHi(i),x);

end


% fzero method
% for i = 1:T
%
% fun = @(y_C)(exp(-((1-xiHat_C*((y_C-
lambdaHat_C)/deltaHat_C)).^(1/xiHat_C))) - RCUORiskHi(i));
```

```
%
% CEUOR(i) = fzero(fun,y0_C)
%
% end
```

File Name: Retailer_Exp_Unf_Rate2_v3_with_optimization_steps.m

```
% Retailer's Expected Unfilled Order Rate =
% Probability of Retailer's Unfilled Order * Retailer's Unfilled
Order Amount
REUOR =
mle_gevd_REUOR(T,n_R,mu_R,sigma_R,lambdaStart_R,deltaStart_R,xiSt
art_R,...
    MSD, MSDF, RCA, WEP, ROL, WIL, RUO, RUOF);

WI = zeros(1,T);                % Wholesaler's Inventory Level
WPR = zeros(1,T);               % Wholesaler's Procurement Rate

% Customer's Expected Unfilled Order Rate =
% Probability of Customer's Unfilled Order * Customer's Unfilled
Order Amount
CEUOR =
mle_gevd_CEUOR(T,n_C,mu_C,sigma_C,lambdaStart_C,deltaStart_C,xiSt
art_C,...
    WSD, WSDF, CCA, REP, COL, RIL, CUO, CUOF);

RI = zeros(1,T);                % Retailer's Inventory Level
SP = zeros(1,T);                % Retailer's Selling Price
CPR = zeros(1,T);               % Customer's Purchase Rate
RSR = zeros(1,T);               % Retailer's Sales Rate
RPR = zeros(1,T);               % Retailer's Procurement Rate
WSR = zeros(1,T);               % Wholesaler's Sales Rate
Retail_Prof = zeros(1,T);       % Retailer's Profit
Retail_Liquid = zeros(1,T);     % Retailer's Working Capital

WI_0 = 2000;                    % Initial Value = 2000 Units
RI_0 = 200;                     % Initial Value = 200 Units

% Linear Programming by optimizing a function of Profit and
Satisfaction
% Constraints
A = zeros(4,3);
b = zeros(4,1);
Aeq = zeros(1,3);
beq = zeros(1,1);

% Decision Variables
v = zeros(1,3);
z = zeros(1,3*T);
```

```
v0 = zeros(1,3);
obj_val = zeros(1,T);

% v(1) = z(i) = SP;
% v(2) = z(i+T) = CPR;
% v(3) = z(i+2*T) = RPR;

% z(3*i-2) = SP(i);
% z(3*i-1) = CPR(i);
% z(3*i) = RPR(i);


for i = 1:T

%%%%%%%%%%%%%%%%
% Optimization %
%%%%%%%%%%%%%%%%

v0(1) = 5;
v0(2) = 50;
v0(3) = 50;

if i == 1;
    % Objective Function
    fun = @(v)(w1*((v(1)-SCC(i))*RSR(i)) + w2*((RI_0+v(3)-
v(2))*RPP+RSR(i)*v(1)-v(3)*SCC(i)))*(-1);
else
    fun = @(v)(w1*((v(1)-SCC(i))*RSR(i)) + w2*((RI_0+v(3)-
v(2)+(sum(RPR-CPR))*RPP+RSR(i)*v(1)-v(3)*SCC(i)))*(-1);
end

% Inequalities Matrices
A(1,1) = S(i);
A(1,2) = 1;
A(2,1) = -1;
A(3,3) = 1;
A(3,2) = -1;
A(4,2) = -1;

b(1) = C(i);
b(2) = -SCC(i);
b(3) = RIC(i) - RI_0;
b(4) = -CEUOR(i);

% Equation Matrices
Aeq(1) = 0;
Aeq(2) = 0;
Aeq(3) = 1;

if i == 1;
    beq(1) = 0;
else
```

```
    if (RI_0 + sum(RPR-CPR)) <= 150;     % if RI(i-1) <= 150
        beq(1) = RSR(i-1) + 50;
    else
        beq(1) = RSR(i-1);
    end
end

% Variables non-negative
lb = [0 0 0];

% Execute Optimization
[z(3*i-2:3*i),fval] = fmincon(fun,v0,A,b,Aeq,beq,lb);

SP(i) = z(3*i-2);
CPR(i) = z(3*i-1);
RPR(i) = z(3*i);
obj_val(i) = -fval;

if i == 1;
    RI(i) = RI_0 + RPR(i) - CPR(i);
else
    RI(i) = RI(i-1) + RPR(i) - CPR(i);
end

%%%%%%%%%%%%%%%%%
% SD Simulation %
%%%%%%%%%%%%%%%%%

% Retailer Sales Rate = Customer Purchase Rate - Customer's
Expected Unfilled Order Rate
RSR(i) = CPR(i) - CEUOR(i);

% Wholesaler Sales Rate = Retailer Procurement Rate -
% Retailer's Expected Unfilled Order Rate
WSR(i) = RPR(i) - REUOR(i);

% Wholesaler Procument Rate Plan
if i == 1;
    WPR(i) = WSR(i);
else
    if (WI_0 + sum(WPR-RPR)) <= 1950;
        WPR(i) = WSR(i-1) + 100;
    else WPR(i) = WSR(i-1);
    end
end

% Wholesaler's Inventory = Wholesaler Procurement Rate - Retailer
Procurement Rate
if i == 1;
    WI(i) = WI_0 + WPR(i) - RPR(i);
else
    WI(i) = WI(i-1) + WPR(i) - RPR(i);
```

```
end

Retail_Prof(i) = (SP(i) - SCC(i)) * RSR(i);
Retail_Liquid(i) = RI(i) * RPP + RSR(i) * SP(i) - RPR(i) *
SCC(i);
end

% Results
WI

RI

Retail_Prof

Retail_Liquid

% Figures
x = 1:T;
figure
plot(x, WI, '--or')
axis([0 T 0 2500])

figure
plot(x, RI, '-*k')
axis([0 T 0 250])

figure
plot(x, obj_val, '-xb')
axis([0 T 0 500])
```

Appendix III –MATLAB Code for the EDFBN Case Study

File Name: Window_Assembly_Data.m

```
% Data of the Assembly Delays DBN and SD

TS = 12; % Simulation running time

FMSPLR   = [0.2 0.8];
FMSPA    = [0.1 0.9];
RFMS     = [0.2 0.8];
FMOSD    = [0.6 0.4];
FMPD     = [0.7 0.5 0.9 0.1 0.85 0.2 0.5 0.05 ...
            0.3 0.5 0.1 0.9 0.15 0.8 0.5 0.95];
FMSD     = [0.8 0.85 0.6 0.8 0.7 0.2 0.4 0.1 ...
            0.2 0.15 0.4 0.2 0.3 0.8 0.6 0.9];
FSFS      = [0.9 0.1];
FSTA      = [0.05 0.95];
WFMSD    = [0.9 0.8 0.15 0.5 0.4 0.8 0.1 0.9 ...
            0.1 0.2 0.85 0.5 0.6 0.2 0.9 0.1];
FAMMD    = [0.1 0.9];
MDDT     = [0.15 0.85];
PFMQ     = [0.2 0.8];
FMQI     = [0.1 0.4 0.4 0.8 ...
            0.9 0.6 0.6 0.2];
FAMF     = [0.5 0.5];
FAMM     = [0.01 0.99];
TFAWS    = [0.3 0.7];
PFAO     = [0.1 0.9];
FAWD     = [0.1 0.3 0.5 0.7 ...
            0.9 0.7 0.5 0.3];
FAMD     = [0.7 0.9 0.75 0.4 0.6 0.2 0.1 0.3 ...
            0.3 0.1 0.25 0.6 0.4 0.8 0.9 0.7];
FAD      = [0.95 0.9 0.6 0.5 0.9 0.1 0.9 0.4 ...
            0.05 0.1 0.4 0.5 0.1 0.9 0.1 0.6];
RGSPLR   = [0.3 0.7];
RGSPA    = [0.05 0.95];
RGMS     = [0.2 0.8];
RGSFRTL  = [0.3 0.7];
RGTA     = [0.2 0.8];
RGSD     = [0.95 0.9 0.85 0.8 0.7 0.6 0.4 0.1 ...
            0.05 0.1 0.15 0.2 0.3 0.4 0.6 0.9];
RGPD     = [0.9 0.7 0.7 0.5 0.5 0.4 0.15 0.05 ...
            0.1 0.3 0.3 0.5 0.5 0.6 0.85 0.95];
RGOSD    = [0.3 0.7];
WRGSD    = [0.9 0.9 0.8 0.6 0.4 0.3 0.2 0.05 ...
            0.1 0.1 0.2 0.4 0.6 0.7 0.8 0.95];
PWAO     = [0.2 0.8];
TWAWS    = [0.05 0.95];
AWI      = [0.9 0.8 0.3 0.1 ...
            0.1 0.2 0.7 0.9];
```

```
WAD      = [0.9 0.7 0.2 0.05 ...
           0.1 0.3 0.8 0.95];
AMI      = [0.95 0.9 0.8 0.8 0.5 0.4 0.3 0.3 ...
           0.05 0.1 0.2 0.2 0.5 0.6 0.7 0.7];
WAMF     = [0.2 0.8];
WAMMD    = [0.3 0.7];
WAMM     = [0.2 0.8];
PWS      = [0.3 0.7];
PWM      = [0.1 0.9];
POD      = [0.9 0.7 0.3 0.1 ...
           0.1 0.3 0.7 0.9];
PMD      = [0.05 0.95];
PPD      = [0.8 0.6 0.4 0.1 ...
           0.2 0.4 0.6 0.9];
DWM      = [0.2 0.8];
DWS      = [0.05 0.95];
SFS      = [0.4 0.6];
PDD      = [0.8 0.6 0.5 0.2 ...
           0.2 0.4 0.5 0.8];
PSD      = [0.7 0.6 0.6 0.3 ...
           0.3 0.4 0.4 0.7];
GDDT     = [0.1 0.9];
GCMF     = [0.1 0.9];
GCMMD    = [0.2 0.8];
GCMM     = [0.2 0.8];
GCWS     = [0.1 0.9];
PGCO     = [0.2 0.8];
GCWD     = [0.8 0.5 0.5 0.3 ...
           0.2 0.5 0.5 0.7];
GCMD     = [0.95 0.9 0.7 0.5 0.5 0.2 0.1 0.05 ...
           0.05 0.1 0.3 0.5 0.5 0.8 0.9 0.95];
CGQI     = [0.9 0.7 0.3 0.1 ...
           0.1 0.3 0.7 0.9];
PGQ      = [0.1 0.9];
GCD      = [0.95 0.9 0.5 0.5 0.3 0.2 0.1 0.05 ...
           0.05 0.1 0.5 0.5 0.7 0.8 0.9 0.95];
FAD_F    = [0.99 0.95 0.95 0.9 0.9 0.8 0.85 0.8 0.7 0.6 0.5 0.5
0.45 0.3 0.1 0.05 ...
           0.01 0.05 0.05 0.1 0.1 0.2 0.15 0.2 0.3 0.4 0.5 0.5
0.55 0.7 0.9 0.95];
WAD_F    = [0.99 0.95 0.9 0.8 0.9 0.8 0.7 0.7 0.5 0.6 0.5 0.5 0.45
0.2 0.15 0.05 ...
           0.01 0.05 0.1 0.2 0.1 0.2 0.3 0.3 0.5 0.4 0.5 0.5 0.55
0.8 0.85 0.95];
PPD_F    = [0.95 0.9 0.7 0.6 0.5 0.4 0.2 0.1 ...
           0.05 0.1 0.3 0.4 0.5 0.6 0.8 0.9];
PSD_F    = [0.9 0.8 0.6 0.4 0.4 0.3 0.2 0.05 ...
           0.1 0.2 0.4 0.6 0.6 0.7 0.8 0.95];
GCD_F    = [0.99 0.95 0.9 0.8 0.9 0.8 0.7 0.7 0.5 0.6 0.5 0.5 0.45
0.2 0.15 0.05 ...
           0.01 0.05 0.1 0.2 0.1 0.2 0.3 0.3 0.5 0.4 0.5 0.5 0.55
0.8 0.85 0.95];
```

```
% VaR Data Generator
n_WFMSD = 1000; mu_WFMSD = 20; sigma_WFMSD = 80;
lambdaStart_WFMSD = -10; deltaStart_WFMSD = 78; xiStart_WFMSD =
0.26;

n_WRGSD = 1000; mu_WRGSD = 10; sigma_WRGSD = 40;
lambdaStart_WRGSD = -5; deltaStart_WRGSD = 40; xiStart_WRGSD =
0.27;

n_WAD = 1000; mu_WAD = 2; sigma_WAD = 6;
lambdaStart_WAD = 0; deltaStart_WAD = 6; xiStart_WAD = 0.26;

n_PSD = 1000; mu_PSD = 10; sigma_PSD = 30;
lambdaStart_PSD = -1; deltaStart_PSD = 30; xiStart_PSD = 0.25;

n_PPD = 1000; mu_PPD = 3; sigma_PPD = 12;
lambdaStart_PPD = -2; deltaStart_PPD = 12; xiStart_PPD = 0.26;

n_GCD = 1000; mu_GCD = 1; sigma_GCD = 2;
lambdaStart_GCD = 0; deltaStart_GCD = 2; xiStart_GCD = 0.35;

n_FAD = 1000; mu_FAD = 5; sigma_FAD = 15;
lambdaStart_FAD = -2; deltaStart_FAD = 15; xiStart_FAD = 0.26;

% Initial Inventory
WFMI_0 = 30;
AWFI_0 = 20;
RGI_0 = 50;
CWGI_0 = 30;
FWI_0 = 150;
PWI_0 = 250;

% Price Curve
demand_0 = 3000; price_slope = 20;

% Marginal Costs
MRMPC = [1.5 10 1;
         2.5 20 1]; % Marginal Raw Material Production Cost
MWAPC = [0.3 0 30;
         0.1 0 20;
         0.5 0 25;
         8.5 0 5;
         4 0 5]; % Mariginal Window Assembly Processing Cost
```

File name: Assembly_Delays.m

```
% Window Assembly Delays Core DBN File

function     [ProbWFMSD, ProbFAD, ProbWRGSD, ProbWAD, ProbPPD,
ProbPSD, ProbGCD] ...
      = Assembly_Delays(FMSPLR, FMSPA, RFMS, FMOSD, FMPD, FMSD,
FSFS, FSTA, ...
                        WFMSD, FAMMD, MDDT, PFMQ, FMQI, FAMF,
FAMM, TFAWS, ...
                        PFAO, FAWD, FAMD, FAD, RGSPLR, RGSPA,
RGMS, RGSFRTL, ...
                        RGTA, RGSD, RGPD, RGOSD, WRGSD, PWAO,
TWAWS, AWI, ...
                        WAD, AMI, WAMF, WAMMD, WAMM, PWS, PWM,
POD, PMD, ...
                        PPD, DWM, DWS, SFS, PDD, PSD, GDDT, GCMF,
GCMMD, ...
                        GCMM, GCWS, PGCO, GCWD, GCMD, CGQI, PGQ,
GCD, ...
                        FAD_F, WAD_F, PPD_F, PSD_F, GCD_F)

% #codegen
ss = 58; % slice size

% Nodes inside one time slice
intra = zeros(ss);
intra(1,5) = 1;
intra(2,5) = 1;
intra(3,5) = 1;
intra(4, 9) = 1;
intra(5, [6 9]) = 1;
intra(6, 9) = 1;
intra(7, 6) = 1;
intra(8, 6) = 1;

intra(10, 19) = 1;
intra(11, 13) = 1;
intra(12, 13) = 1;
intra(13, 20) = 1;
intra(14, 19) = 1;
intra(15, 19) = 1;
intra(16, 18) = 1;
intra(17, 18) = 1;
intra(18, 20) = 1;
intra(19, 20) = 1;

intra(21, 27) = 1;
intra(22, 27) = 1;
intra(23, 27) = 1;
intra(24, 26) = 1;
```

```
intra(25, 26) = 1;
intra(26, 29) = 1;
intra(27, [26 29]) = 1;
intra(28, 29) = 1;

intra(30, 32) = 1;
intra(31, 32) = 1;
intra(32, 33) = 1;

intra(34, 33) = 1;
intra(35, 34) = 1;
intra(36, 34) = 1;
intra(37, 34) = 1;
intra(38, 40) = 1;
intra(39, 40) = 1;
intra(40, 42) = 1;
intra(41, 42) = 1;

intra(43, 46) = 1;
intra(44, 46) = 1;
intra(45, 47) = 1;
intra(46, 47) = 1;

intra(48, 56) = 1;
intra(49, 55) = 1;
intra(50, 55) = 1;
intra(51, 55) = 1;
intra(52, 54) = 1;
intra(53, 54) = 1;
intra(54, 58) = 1;
intra(55, 58) = 1;
intra(56, 58) = 1;
intra(57, 56) = 1;

% Nodes with inter-time-slice interaction
inter = zeros(ss);
inter(9, 20) = 1;
inter(20, 33) = 1;
inter(29, 58) = 1;
inter(33, 42) = 1;
inter(42, 47) = 1;
inter(58, 33) = 1;

% onodes = [2 10 30]; % observed
dnodes = 1:ss; % discrete
ns = 2*ones(1,ss); % binary nodes

TS = 12; % Number of Time Slice

% Define Temporal Probability Values for Output Variables
Occuring
ProbWFMSD = zeros(1,TS);
```

```
ProbFAD = zeros(1,TS);
ProbWRGSD = zeros(1,TS);
ProbWAD = zeros(1,TS);
ProbPPD = zeros(1,TS);
ProbPSD = zeros(1,TS);
ProbGCD = zeros(1,TS);



% DBN (In order to get the probability value at t = 2:TS)
eclass1 = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
22 23 24 25 26 27 28 ...
    29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
49 50 51 52 53 54 55 56 57 58];
eclass2 = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 59 21
22 23 24 25 26 27 28 ...
    29 30 31 32 60 34 35 36 37 38 39 40 41 61 43 44 45 46 62 48
49 50 51 52 53 54 55 56 57 63];
% eclass = [eclass1 eclass2];

onodes = [2 7 10 14 23 25 30 38 44 51 57];

bnet2 = mk_dbn(intra, inter, ns, 'discrete', dnodes, 'observed',
onodes, 'eclass1', eclass1, 'eclass2', eclass2);

bnet2.CPD{1} = tabular_CPD(bnet2, 1, FMSPLR);
bnet2.CPD{2} = tabular_CPD(bnet2, 2, FMSPA);
bnet2.CPD{3} = tabular_CPD(bnet2, 3, RFMS);
bnet2.CPD{4} = tabular_CPD(bnet2, 4, FMOSD);
bnet2.CPD{5} = tabular_CPD(bnet2, 5, FMPD);
bnet2.CPD{6} = tabular_CPD(bnet2, 6, FMSD);
bnet2.CPD{7} = tabular_CPD(bnet2, 7, FSFS);
bnet2.CPD{8} = tabular_CPD(bnet2, 8, FSTA);
bnet2.CPD{9} = tabular_CPD(bnet2, 9, WFMSD);
bnet2.CPD{10} = tabular_CPD(bnet2, 10, FAMMD);
bnet2.CPD{11} = tabular_CPD(bnet2, 11, MDDT);
bnet2.CPD{12} = tabular_CPD(bnet2, 12, PFMQ);
bnet2.CPD{13} = tabular_CPD(bnet2, 13, FMQI);
bnet2.CPD{14} = tabular_CPD(bnet2, 14, FAMF);
bnet2.CPD{15} = tabular_CPD(bnet2, 15, FAMM);
bnet2.CPD{16} = tabular_CPD(bnet2, 16, TFAWS);
bnet2.CPD{17} = tabular_CPD(bnet2, 17, PFAO);
bnet2.CPD{18} = tabular_CPD(bnet2, 18, FAWD);
bnet2.CPD{19} = tabular_CPD(bnet2, 19, FAMD);
bnet2.CPD{20} = tabular_CPD(bnet2, 20, FAD);
bnet2.CPD{21} = tabular_CPD(bnet2, 21, RGSPLR);
bnet2.CPD{22} = tabular_CPD(bnet2, 22, RGSPA);
bnet2.CPD{23} = tabular_CPD(bnet2, 23, RGMS);
bnet2.CPD{24} = tabular_CPD(bnet2, 24, RGSFRTL);
bnet2.CPD{25} = tabular_CPD(bnet2, 25, RGTA);
bnet2.CPD{26} = tabular_CPD(bnet2, 26, RGSD);
bnet2.CPD{27} = tabular_CPD(bnet2, 27, RGPD);
```

```
bnet2.CPD{28} = tabular_CPD(bnet2, 28, RGOSD);
bnet2.CPD{29} = tabular_CPD(bnet2, 29, WRGSD);
bnet2.CPD{30} = tabular_CPD(bnet2, 30, PWAO);
bnet2.CPD{31} = tabular_CPD(bnet2, 31, TWAWS);
bnet2.CPD{32} = tabular_CPD(bnet2, 32, AWI);
bnet2.CPD{33} = tabular_CPD(bnet2, 33, WAD);
bnet2.CPD{34} = tabular_CPD(bnet2, 34, AMI);
bnet2.CPD{35} = tabular_CPD(bnet2, 35, WAMF);
bnet2.CPD{36} = tabular_CPD(bnet2, 36, WAMMD);
bnet2.CPD{37} = tabular_CPD(bnet2, 37, WAMM);
bnet2.CPD{38} = tabular_CPD(bnet2, 38, PWS);
bnet2.CPD{39} = tabular_CPD(bnet2, 39, PWM);
bnet2.CPD{40} = tabular_CPD(bnet2, 40, POD);
bnet2.CPD{41} = tabular_CPD(bnet2, 41, PMD);
bnet2.CPD{42} = tabular_CPD(bnet2, 42, PPD);
bnet2.CPD{43} = tabular_CPD(bnet2, 43, DWM);
bnet2.CPD{44} = tabular_CPD(bnet2, 44, DWS);
bnet2.CPD{45} = tabular_CPD(bnet2, 45, SFS);
bnet2.CPD{46} = tabular_CPD(bnet2, 46, PDD);
bnet2.CPD{47} = tabular_CPD(bnet2, 47, PSD);
bnet2.CPD{48} = tabular_CPD(bnet2, 48, GDDT);
bnet2.CPD{49} = tabular_CPD(bnet2, 49, GCMF);
bnet2.CPD{50} = tabular_CPD(bnet2, 50, GCMMD);
bnet2.CPD{51} = tabular_CPD(bnet2, 51, GCMM);
bnet2.CPD{52} = tabular_CPD(bnet2, 52, GCWS);
bnet2.CPD{53} = tabular_CPD(bnet2, 53, PGCO);
bnet2.CPD{54} = tabular_CPD(bnet2, 54, GCWD);
bnet2.CPD{55} = tabular_CPD(bnet2, 55, GCMD);
bnet2.CPD{56} = tabular_CPD(bnet2, 56, CGQI);
bnet2.CPD{57} = tabular_CPD(bnet2, 57, PGQ);
bnet2.CPD{58} = tabular_CPD(bnet2, 58, GCD);
bnet2.CPD{59} = tabular_CPD(bnet2, 78, FAD_F);
bnet2.CPD{60} = tabular_CPD(bnet2, 91, WAD_F);
bnet2.CPD{61} = tabular_CPD(bnet2, 100, PPD_F);
bnet2.CPD{62} = tabular_CPD(bnet2, 105, PSD_F);
bnet2.CPD{63} = tabular_CPD(bnet2, 116, GCD_F);


ev = sample_dbn(bnet2, TS);
evidence2 = cell(ss,TS);

evidence2(onodes,:) = ev(onodes,:); % all cells besides onodes
are empty

for T = [1 10]
    evidence2{onodes(1),T} = 1;
end

for T = [2 11]
    evidence2{onodes(1),T} = 2;
end
```

```
for T = [3 7]
    evidence2{onodes(2),T} = 1;
end

for T = [1 4 7 10]
    evidence2{onodes(3),T} = 1;
end

for T = [2 5 8 11]
    evidence2{onodes(3),T} = 2;
end

for T = [2 9]
    evidence2{onodes(4),T} = 1;
end

for T = 5
    evidence2{onodes(5),T} = 1;
end

for T = [3 6]
    evidence2{onodes(6),T} = 1;
end

for T = [1 3 5]
    evidence2{onodes(7),T} = 1;
end

for T = 6
    evidence2{onodes(7),T} = 2;
end

for T = [7 8 9]
    evidence2{onodes(8),T} = 1;
end

for T = 10
    evidence2{onodes(8),T} = 2;
end

for T = [7 8 9]
    evidence2{onodes(9),T} = 1;
end

for T = 10
    evidence2{onodes(9),T} = 2;
end

for T = [3 9]
    evidence2{onodes(10),T} = 1;
end
```

```
for T = 5
    evidence2{onodes(11),T} = 1;
end

engine2 = smoother_engine(jtree_2TBN_inf_engine(bnet2));
engine2 = enter_evidence(engine2, evidence2);


for T = 1:TS

m9 = marginal_nodes(engine2, 9, T);
m20 = marginal_nodes(engine2, 20, T);
m29 = marginal_nodes(engine2, 29, T);
m33 = marginal_nodes(engine2, 33, T);
m42 = marginal_nodes(engine2, 42, T);
m47 = marginal_nodes(engine2, 47, T);
m58 = marginal_nodes(engine2, 58, T);

% marg.T    % Inference Answer
ProbWFMSD(T)     = m9.T(1);
ProbFAD(T)       = m20.T(1);
ProbWRGSD(T)     = m29.T(1);
ProbWAD(T)       = m33.T(1);
ProbPPD(T)       = m42.T(1);
ProbPSD(T)       = m47.T(1);
ProbGCD(T)       = m58.T(1);

end


% Illustration

% isbox = zeros(ss,1); isbox(dnodes) = 1;
% unfold = 4;
% draw_dbn(intra, inter, 0, unfold, {'a', 'b', 'c', 'd', 'e',
'f'}, isbox); % Draw DBN Diagram

% ProbCustOrdLvl = ProbCustOrdLvl.VRiskHi;
```

File Name: Assembly_Delays_no_miti.m

```
% Window Assembly Delays Core DBN File

function    [ProbWFMSD, ProbFAD, ProbWRGSD, ProbWAD, ProbPPD,
ProbPSD, ProbGCD] ...
```

```
      = Assembly_Delays(FMSPLR, FMSPA, RFMS, FMOSD, FMPD, FMSD,
FSFS, FSTA, ...
                        WFMSD, FAMMD, MDDT, PFMQ, FMQI, FAMF,
FAMM, TFAWS, ...
                        PFAO, FAWD, FAMD, FAD, RGSPLR, RGSPA,
RGMS, RGSFRTL, ...
                        RGTA, RGSD, RGPD, RGOSD, WRGSD, PWAO,
TWAWS, AWI, ...
                        WAD, AMI, WAMF, WAMMD, WAMM, PWS, PWM,
POD, PMD, ...
                        PPD, DWM, DWS, SFS, PDD, PSD, GDDT, GCMF,
GCMMD, ...
                        GCMM, GCWS, PGCO, GCWD, GCMD, CGQI, PGQ,
GCD, ...
                        FAD_F, WAD_F, PPD_F, PSD_F, GCD_F)

% #codegen
ss = 58; % slice size

% Nodes inside one time slice
intra = zeros(ss);
intra(1,5) = 1;
intra(2,5) = 1;
intra(3,5) = 1;
intra(4, 9) = 1;
intra(5, [6 9]) = 1;
intra(6, 9) = 1;
intra(7, 6) = 1;
intra(8, 6) = 1;

intra(10, 19) = 1;
intra(11, 13) = 1;
intra(12, 13) = 1;
intra(13, 20) = 1;
intra(14, 19) = 1;
intra(15, 19) = 1;
intra(16, 18) = 1;
intra(17, 18) = 1;
intra(18, 20) = 1;
intra(19, 20) = 1;

intra(21, 27) = 1;
intra(22, 27) = 1;
intra(23, 27) = 1;
intra(24, 26) = 1;
intra(25, 26) = 1;
intra(26, 29) = 1;
intra(27, [26 29]) = 1;
intra(28, 29) = 1;

intra(30, 32) = 1;
intra(31, 32) = 1;
```

```
intra(32, 33) = 1;

intra(34, 33) = 1;
intra(35, 34) = 1;
intra(36, 34) = 1;
intra(37, 34) = 1;
intra(38, 40) = 1;
intra(39, 40) = 1;
intra(40, 42) = 1;
intra(41, 42) = 1;

intra(43, 46) = 1;
intra(44, 46) = 1;
intra(45, 47) = 1;
intra(46, 47) = 1;

intra(48, 56) = 1;
intra(49, 55) = 1;
intra(50, 55) = 1;
intra(51, 55) = 1;
intra(52, 54) = 1;
intra(53, 54) = 1;
intra(54, 58) = 1;
intra(55, 58) = 1;
intra(56, 58) = 1;
intra(57, 56) = 1;

% Nodes with inter-time-slice interaction
inter = zeros(ss);
inter(9, 20) = 1;
inter(20, 33) = 1;
inter(29, 58) = 1;
inter(33, 42) = 1;
inter(42, 47) = 1;
inter(58, 33) = 1;

% onodes = [2 10 30]; % observed
dnodes = 1:ss; % discrete
ns = 2*ones(1,ss); % binary nodes

TS = 12; % Number of Time Slice

% Define Temporal Probability Values for Output Variables
Occuring
ProbWFMSD = zeros(1,TS);
ProbFAD = zeros(1,TS);
ProbWRGSD = zeros(1,TS);
ProbWAD = zeros(1,TS);
ProbPPD = zeros(1,TS);
ProbPSD = zeros(1,TS);
ProbGCD = zeros(1,TS);
```

```
% DBN (In order to get the probability value at t = 2:TS)
eclass1 = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
22 23 24 25 26 27 28 ...
    29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
49 50 51 52 53 54 55 56 57 58];
eclass2 = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 59 21
22 23 24 25 26 27 28 ...
    29 30 31 32 60 34 35 36 37 38 39 40 41 61 43 44 45 46 62 48
49 50 51 52 53 54 55 56 57 63];
% eclass = [eclass1 eclass2];

onodes = [2 7 10 14 23 25 30 38 44 51 57];

bnet2 = mk_dbn(intra, inter, ns, 'discrete', dnodes, 'observed',
onodes, 'eclass1', eclass1, 'eclass2', eclass2);

bnet2.CPD{1} = tabular_CPD(bnet2, 1, FMSPLR);
bnet2.CPD{2} = tabular_CPD(bnet2, 2, FMSPA);
bnet2.CPD{3} = tabular_CPD(bnet2, 3, RFMS);
bnet2.CPD{4} = tabular_CPD(bnet2, 4, FMOSD);
bnet2.CPD{5} = tabular_CPD(bnet2, 5, FMPD);
bnet2.CPD{6} = tabular_CPD(bnet2, 6, FMSD);
bnet2.CPD{7} = tabular_CPD(bnet2, 7, FSFS);
bnet2.CPD{8} = tabular_CPD(bnet2, 8, FSTA);
bnet2.CPD{9} = tabular_CPD(bnet2, 9, WFMSD);
bnet2.CPD{10} = tabular_CPD(bnet2, 10, FAMMD);
bnet2.CPD{11} = tabular_CPD(bnet2, 11, MDDT);
bnet2.CPD{12} = tabular_CPD(bnet2, 12, PFMQ);
bnet2.CPD{13} = tabular_CPD(bnet2, 13, FMQI);
bnet2.CPD{14} = tabular_CPD(bnet2, 14, FAMF);
bnet2.CPD{15} = tabular_CPD(bnet2, 15, FAMM);
bnet2.CPD{16} = tabular_CPD(bnet2, 16, TFAWS);
bnet2.CPD{17} = tabular_CPD(bnet2, 17, PFAO);
bnet2.CPD{18} = tabular_CPD(bnet2, 18, FAWD);
bnet2.CPD{19} = tabular_CPD(bnet2, 19, FAMD);
bnet2.CPD{20} = tabular_CPD(bnet2, 20, FAD);
bnet2.CPD{21} = tabular_CPD(bnet2, 21, RGSPLR);
bnet2.CPD{22} = tabular_CPD(bnet2, 22, RGSPA);
bnet2.CPD{23} = tabular_CPD(bnet2, 23, RGMS);
bnet2.CPD{24} = tabular_CPD(bnet2, 24, RGSFRTL);
bnet2.CPD{25} = tabular_CPD(bnet2, 25, RGTA);
bnet2.CPD{26} = tabular_CPD(bnet2, 26, RGSD);
bnet2.CPD{27} = tabular_CPD(bnet2, 27, RGPD);
bnet2.CPD{28} = tabular_CPD(bnet2, 28, RGOSD);
bnet2.CPD{29} = tabular_CPD(bnet2, 29, WRGSD);
bnet2.CPD{30} = tabular_CPD(bnet2, 30, PWAO);
bnet2.CPD{31} = tabular_CPD(bnet2, 31, TWAWS);
bnet2.CPD{32} = tabular_CPD(bnet2, 32, AWI);
bnet2.CPD{33} = tabular_CPD(bnet2, 33, WAD);
bnet2.CPD{34} = tabular_CPD(bnet2, 34, AMI);
```

```
bnet2.CPD{35} = tabular_CPD(bnet2, 35, WAMF);
bnet2.CPD{36} = tabular_CPD(bnet2, 36, WAMMD);
bnet2.CPD{37} = tabular_CPD(bnet2, 37, WAMM);
bnet2.CPD{38} = tabular_CPD(bnet2, 38, PWS);
bnet2.CPD{39} = tabular_CPD(bnet2, 39, PWM);
bnet2.CPD{40} = tabular_CPD(bnet2, 40, POD);
bnet2.CPD{41} = tabular_CPD(bnet2, 41, PMD);
bnet2.CPD{42} = tabular_CPD(bnet2, 42, PPD);
bnet2.CPD{43} = tabular_CPD(bnet2, 43, DWM);
bnet2.CPD{44} = tabular_CPD(bnet2, 44, DWS);
bnet2.CPD{45} = tabular_CPD(bnet2, 45, SFS);
bnet2.CPD{46} = tabular_CPD(bnet2, 46, PDD);
bnet2.CPD{47} = tabular_CPD(bnet2, 47, PSD);
bnet2.CPD{48} = tabular_CPD(bnet2, 48, GDDT);
bnet2.CPD{49} = tabular_CPD(bnet2, 49, GCMF);
bnet2.CPD{50} = tabular_CPD(bnet2, 50, GCMMD);
bnet2.CPD{51} = tabular_CPD(bnet2, 51, GCMM);
bnet2.CPD{52} = tabular_CPD(bnet2, 52, GCWS);
bnet2.CPD{53} = tabular_CPD(bnet2, 53, PGCO);
bnet2.CPD{54} = tabular_CPD(bnet2, 54, GCWD);
bnet2.CPD{55} = tabular_CPD(bnet2, 55, GCMD);
bnet2.CPD{56} = tabular_CPD(bnet2, 56, CGQI);
bnet2.CPD{57} = tabular_CPD(bnet2, 57, PGQ);
bnet2.CPD{58} = tabular_CPD(bnet2, 58, GCD);
bnet2.CPD{59} = tabular_CPD(bnet2, 78, FAD_F);
bnet2.CPD{60} = tabular_CPD(bnet2, 91, WAD_F);
bnet2.CPD{61} = tabular_CPD(bnet2, 100, PPD_F);
bnet2.CPD{62} = tabular_CPD(bnet2, 105, PSD_F);
bnet2.CPD{63} = tabular_CPD(bnet2, 116, GCD_F);


ev = sample_dbn(bnet2, TS);
evidence2 = cell(ss,TS);

evidence2(onodes,:) = ev(onodes,:); % all cells besides onodes
are empty

for T = [1 10]
    evidence2{onodes(1),T} = 1;
end

% for T = [2 11]
%     evidence2{onodes(1),T} = 2;
% end

for T = [3 7]
    evidence2{onodes(2),T} = 1;
end

for T = [1 4 7 10]
    evidence2{onodes(3),T} = 1;
end
```

```
% for T = [2 5 8 11]
%     evidence2{onodes(3),T} = 2;
% end

for T = [2 9]
    evidence2{onodes(4),T} = 1;
end

for T = 5
    evidence2{onodes(5),T} = 1;
end

for T = [3 6]
    evidence2{onodes(6),T} = 1;
end

for T = [1 3 5]
    evidence2{onodes(7),T} = 1;
end

% for T = 6
%     evidence2{onodes(7),T} = 2;
% end

for T = [7 8 9]
    evidence2{onodes(8),T} = 1;
end

% for T = 10
%     evidence2{onodes(8),T} = 2;
% end

for T = [7 8 9]
    evidence2{onodes(9),T} = 1;
end

% for T = 10
%     evidence2{onodes(9),T} = 2;
% end

for T = [3 9]
    evidence2{onodes(10),T} = 1;
end

for T = 5
    evidence2{onodes(11),T} = 1;
end

engine2 = smoother_engine(jtree_2TBN_inf_engine(bnet2));
engine2 = enter_evidence(engine2, evidence2);
```

```
for T = 1:TS

m9 = marginal_nodes(engine2, 9, T);
m20 = marginal_nodes(engine2, 20, T);
m29 = marginal_nodes(engine2, 29, T);
m33 = marginal_nodes(engine2, 33, T);
m42 = marginal_nodes(engine2, 42, T);
m47 = marginal_nodes(engine2, 47, T);
m58 = marginal_nodes(engine2, 58, T);

% marg.T    % Inference Answer
ProbWFMSD(T)     = m9.T(1);
ProbFAD(T)       = m20.T(1);
ProbWRGSD(T)     = m29.T(1);
ProbWAD(T)       = m33.T(1);
ProbPPD(T)       = m42.T(1);
ProbPSD(T)       = m47.T(1);
ProbGCD(T)       = m58.T(1);

end


% Illustration

% isbox = zeros(ss,1); isbox(dnodes) = 1;
% unfold = 4;
% draw_dbn(intra, inter, 0, unfold, {'a', 'b', 'c', 'd', 'e',
'f'}, isbox); % Draw DBN Diagram

% ProbCustOrdLvl = ProbCustOrdLvl.VRiskHi;
```

File Name: mle_gevd_frame_material.m

```
% Generalized extreme value distributions (GEVD) Parameter
Estimate for
% Window Frame Materials Supplying Delay

x_gen_WFMSD = normrnd(mu_WFMSD, sigma_WFMSD, n_WFMSD, 1);

x_gen_WFMSD2 = zeros(n_WFMSD,1);

for i = 1:n_WFMSD

    if x_gen_WFMSD(i) >= 0;
        x_gen_WFMSD2(i) = x_gen_WFMSD(i);
```

```
    else x_gen_WFMSD2(i) = 0;
    end

end

pdf_gevd_WFMSD = @(x,lambda,delta,xi) ...
    (1./delta).*(exp(-(1-xi.*((x-lambda)./delta)).^(1./xi))).*(1-
xi.*((x-lambda)./delta)).^((1./xi)-1);

start = [lambdaStart_WFMSD deltaStart_WFMSD xiStart_WFMSD];

lb = [];
ub = [];
gevd_paramEsts_WFMSD = mle(x_gen_WFMSD, 'pdf',pdf_gevd_WFMSD,
'start',start, ...
    'lower',lb, 'upper',ub);

lambdaHat_WFMSD = gevd_paramEsts_WFMSD(1);
deltaHat_WFMSD = gevd_paramEsts_WFMSD(2);
xiHat_WFMSD = gevd_paramEsts_WFMSD(3);

% lambdaHat_C = round((gevd_paramEsts(1)*10000))/10000;
% deltaHat_C = round((gevd_paramEsts(1)*10000))/10000;
% xiHat_C = round((gevd_paramEsts(1)*10000))/10000;

ProbWFMSD = Assembly_Delays(FMSPLR, FMSPA, RFMS, FMOSD, FMPD,
FMSD, FSFS, FSTA, ...
                        WFMSD, FAMMD, MDDT, PFMQ, FMQI, FAMF,
FAMM, TFAWS, ...
                        PFAO, FAWD, FAMD, FAD, RGSPLR, RGSPA,
RGMS, RGSFRTL, ...
                        RGTA, RGSD, RGPD, RGOSD, WRGSD, PWAO,
TWAWS, AWI, ...
                        WAD, AMI, WAMF, WAMMD, WAMM, PWS, PWM,
POD, PMD, ...
                        PPD, DWM, DWS, SFS, PDD, PSD, GDDT, GCMF,
GCMMD, ...
                        GCMM, GCWS, PGCO, GCWD, GCMD, CGQI, PGQ,
GCD, ...
                        FAD_F, WAD_F, PPD_F, PSD_F, GCD_F);

WFMSD_Amount = zeros(1,TS);

syms x

cdf_gevd_WFMSD(x) = exp(-(1-xiHat_WFMSD.*((x-
lambdaHat_WFMSD)./deltaHat_WFMSD)).^(1./xiHat_WFMSD));


for i = 1:TS

WFMSD_Amount(i) = solve(cdf_gevd_WFMSD == ProbWFMSD(i),x);
```

```
end


% fzero method
% for i = 1:T
%
% fun = @(y_C)(exp(-((1-xiHat_C*((y_C-
lambdaHat_C)/deltaHat_C)).^(1/xiHat_C))) - RCUORiskHi(i));
%
% CEUOR(i) = fzero(fun,y0_C)
%
% end
```

File Name: mle_gevd_raw_glass.m

```
% Generalized extreme value distributions (GEVD) Parameter
Estimate for
% Window Raw Glass Supplying Delay


x_gen_WRGSD = normrnd(mu_WRGSD, sigma_WRGSD, n_WRGSD, 1);

x_gen_WRGSD2 = zeros(n_WRGSD,1);

for i = 1:n_WRGSD

    if x_gen_WRGSD(i) >= 0;
        x_gen_WRGSD2(i) = x_gen_WRGSD(i);
    else x_gen_WRGSD2(i) = 0;
    end

end

pdf_gevd_WRGSD = @(x,lambda,delta,xi) ...
    (1./delta).*(exp(-(1-xi.*((x-lambda)./delta)).^(1./xi))).*(1-
xi.*((x-lambda)./delta)).^((1./xi)-1);

start = [lambdaStart_WRGSD deltaStart_WRGSD xiStart_WRGSD];

lb = [];
ub = [];
gevd_paramEsts_WRGSD = mle(x_gen_WRGSD, 'pdf',pdf_gevd_WRGSD,
'start',start, ...
    'lower',lb, 'upper',ub);
```

```
lambdaHat_WRGSD = gevd_paramEsts_WRGSD(1);
deltaHat_WRGSD = gevd_paramEsts_WRGSD(2);
xiHat_WRGSD = gevd_paramEsts_WRGSD(3);

% lambdaHat_C = round((gevd_paramEsts(1)*10000))/10000;
% deltaHat_C = round((gevd_paramEsts(1)*10000))/10000;
% xiHat_C = round((gevd_paramEsts(1)*10000))/10000;

ProbWRGSD = Assembly_Delays(FMSPLR, FMSPA, RFMS, FMOSD, FMPD,
FMSD, FSFS, FSTA, ...
                        WFMSD, FAMMD, MDDT, PFMQ, FMQI, FAMF,
FAMM, TFAWS, ...
                        PFAO, FAWD, FAMD, FAD, RGSPLR, RGSPA,
RGMS, RGSFRTL, ...
                        RGTA, RGSD, RGPD, RGOSD, WRGSD, PWAO,
TWAWS, AWI, ...
                        WAD, AMI, WAMF, WAMMD, WAMM, PWS, PWM,
POD, PMD, ...
                        PPD, DWM, DWS, SFS, PDD, PSD, GDDT, GCMF,
GCMMD, ...
                        GCMM, GCWS, PGCO, GCWD, GCMD, CGQI, PGQ,
GCD, ...
                        FAD_F, WAD_F, PPD_F, PSD_F, GCD_F);

WRGSD_Amount = zeros(1,TS);

syms x

cdf_gevd_WRGSD(x) = exp(-(1-xiHat_WRGSD.*((x-
lambdaHat_WRGSD)./deltaHat_WRGSD)).^(1./xiHat_WRGSD));


for i = 1:TS

WRGSD_Amount(i) = solve(cdf_gevd_WRGSD == ProbWRGSD(i),x);

end


% fzero method
% for i = 1:T
%
% fun = @(y_C)(exp(-((1-xiHat_C*((y_C-
lambdaHat_C)/deltaHat_C)).^(1/xiHat_C))) - RCUORiskHi(i));
%
% CEUOR(i) = fzero(fun,y0_C)
%
% end
```

File Name: mle_gevd_glass_cutting.m

```
% Generalized extreme value distributions (GEVD) Parameter
Estimate for
% Window Glass Cutting Delay


x_gen_GCD = normrnd(mu_GCD, sigma_GCD, n_GCD, 1);

x_gen_GCD2 = zeros(n_GCD,1);

for i = 1:n_GCD

    if x_gen_GCD(i) >= 0;
        x_gen_GCD2(i) = x_gen_GCD(i);
    else x_gen_GCD2(i) = 0;
    end

end

pdf_gevd_GCD = @(x,lambda,delta,xi) ...
    (1./delta).*(exp(-(1-xi.*((x-lambda)./delta)).^(1./xi))).*(1-
xi.*((x-lambda)./delta)).^((1./xi)-1);

start = [lambdaStart_GCD deltaStart_GCD xiStart_GCD];

lb = [];
ub = [];
gevd_paramEsts_GCD = mle(x_gen_GCD, 'pdf',pdf_gevd_GCD,
'start',start, ...
    'lower',lb, 'upper',ub);

lambdaHat_GCD = gevd_paramEsts_GCD(1);
deltaHat_GCD = gevd_paramEsts_GCD(2);
xiHat_GCD = gevd_paramEsts_GCD(3);

% lambdaHat_C = round((gevd_paramEsts(1)*10000))/10000;
% deltaHat_C = round((gevd_paramEsts(1)*10000))/10000;
% xiHat_C = round((gevd_paramEsts(1)*10000))/10000;

ProbGCD = Assembly_Delays(FMSPLR, FMSPA, RFMS, FMOSD, FMPD, FMSD,
FSFS, FSTA, ...
                        WFMSD, FAMMD, MDDT, PFMQ, FMQI, FAMF,
FAMM, TFAWS, ...
                        PFAO, FAWD, FAMD, FAD, RGSPLR, RGSPA,
RGMS, RGSFRTL, ...
                        RGTA, RGSD, RGPD, RGOSD, WRGSD, PWAO,
TWAWS, AWI, ...
                        WAD, AMI, WAMF, WAMMD, WAMM, PWS, PWM,
POD, PMD, ...
```

```
                              PPD, DWM, DWS, SFS, PDD, PSD, GDDT, GCMF,
GCMMD, ...
                              GCMM, GCWS, PGCO, GCWD, GCMD, CGQI, PGQ,
GCD, ...
                              FAD_F, WAD_F, PPD_F, PSD_F, GCD_F);

GCD_Amount = zeros(1,TS);

syms x

cdf_gevd_GCD(x) = exp(-(1-xiHat_GCD.*((x-
lambdaHat_GCD)./deltaHat_GCD)).^(1./xiHat_GCD));


for i = 1:TS

GCD_Amount(i) = solve(cdf_gevd_GCD == ProbGCD(i),x);

end


% fzero method
% for i = 1:T
%
% fun = @(y_C)(exp(-((1-xiHat_C*((y_C-
lambdaHat_C)/deltaHat_C)).^(1/xiHat_C))) - RCUORiskHi(i));
%
% CEUOR(i) = fzero(fun,y0_C)
%
% end
```

File Name: mle_gevd_frame_assembly.m

```
% Generalized extreme value distributions (GEVD) Parameter
Estimate for
% Window Frame Assembly Delay

x_gen_FAD = normrnd(mu_FAD, sigma_FAD, n_FAD, 1);

x_gen_FAD2 = zeros(n_FAD,1);

for i = 1:n_FAD

    if x_gen_FAD(i) >= 0;
        x_gen_FAD2(i) = x_gen_FAD(i);
    else x_gen_FAD2(i) = 0;
```

```
    end

end

pdf_gevd_FAD = @(x,lambda,delta,xi) ...
    (1./delta).*(exp(-(1-xi.*((x-lambda)./delta)).^(1./xi))).*(1-
xi.*((x-lambda)./delta)).^((1./xi)-1);

start = [lambdaStart_FAD deltaStart_FAD xiStart_FAD];

lb = [];
ub = [];
gevd_paramEsts_FAD = mle(x_gen_FAD, 'pdf',pdf_gevd_FAD,
'start',start, ...
    'lower',lb, 'upper',ub);

lambdaHat_FAD = gevd_paramEsts_FAD(1);
deltaHat_FAD = gevd_paramEsts_FAD(2);
xiHat_FAD = gevd_paramEsts_FAD(3);

% lambdaHat_C = round((gevd_paramEsts(1)*10000))/10000;
% deltaHat_C = round((gevd_paramEsts(1)*10000))/10000;
% xiHat_C = round((gevd_paramEsts(1)*10000))/10000;

ProbFAD = Assembly_Delays(FMSPLR, FMSPA, RFMS, FMOSD, FMPD, FMSD,
FSFS, FSTA, ...
                        WFMSD, FAMMD, MDDT, PFMQ, FMQI, FAMF,
FAMM, TFAWS, ...
                        PFAO, FAWD, FAMD, FAD, RGSPLR, RGSPA,
RGMS, RGSFRTL, ...
                        RGTA, RGSD, RGPD, RGOSD, WRGSD, PWAO,
TWAWS, AWI, ...
                        WAD, AMI, WAMF, WAMMD, WAMM, PWS, PWM,
POD, PMD, ...
                        PPD, DWM, DWS, SFS, PDD, PSD, GDDT, GCMF,
GCMMD, ...
                        GCMM, GCWS, PGCO, GCWD, GCMD, CGQI, PGQ,
GCD, ...
                        FAD_F, WAD_F, PPD_F, PSD_F, GCD_F);

FAD_Amount = zeros(1,TS);

syms x

cdf_gevd_FAD(x) = exp(-(1-xiHat_FAD.*((x-
lambdaHat_FAD)./deltaHat_FAD)).^(1./xiHat_FAD));


for i = 1:TS

FAD_Amount(i) = solve(cdf_gevd_FAD == ProbFAD(i),x);
```

```
end


% fzero method
% for i = 1:T
%
% fun = @(y_C)(exp(-((1-xiHat_C*((y_C-
lambdaHat_C)/deltaHat_C)).^(1/xiHat_C))) - RCUORiskHi(i));
%
% CEUOR(i) = fzero(fun,y0_C)
%
% end
```

File Name: mle_gevd_window_assembly.m

```
% Generalized extreme value distributions (GEVD) Parameter
Estimate for
% Window Assembly Delay

x_gen_WAD = normrnd(mu_WAD, sigma_WAD, n_WAD, 1);

x_gen_WAD2 = zeros(n_WAD,1);

for i = 1:n_WAD

    if x_gen_WAD(i) >= 0;
        x_gen_WAD2(i) = x_gen_WAD(i);
    else x_gen_WAD2(i) = 0;
    end

end

pdf_gevd_WAD = @(x,lambda,delta,xi) ...
    (1./delta).*(exp(-(1-xi.*((x-lambda)./delta)).^(1./xi))).*(1-
xi.*((x-lambda)./delta)).^((1./xi)-1);

start = [lambdaStart_WAD deltaStart_WAD xiStart_WAD];

lb = [];
ub = [];
gevd_paramEsts_WAD = mle(x_gen_WAD, 'pdf',pdf_gevd_WAD,
'start',start, ...
    'lower',lb, 'upper',ub);

lambdaHat_WAD = gevd_paramEsts_WAD(1);
deltaHat_WAD = gevd_paramEsts_WAD(2);
```

```
xiHat_WAD = gevd_paramEsts_WAD(3);

% lambdaHat_C = round((gevd_paramEsts(1)*10000))/10000;
% deltaHat_C = round((gevd_paramEsts(1)*10000))/10000;
% xiHat_C = round((gevd_paramEsts(1)*10000))/10000;

ProbWAD = Assembly_Delays(FMSPLR, FMSPA, RFMS, FMOSD, FMPD, FMSD,
FSFS, FSTA, ...
                        WFMSD, FAMMD, MDDT, PFMQ, FMQI, FAMF,
FAMM, TFAWS, ...
                        PFAO, FAWD, FAMD, FAD, RGSPLR, RGSPA,
RGMS, RGSFRTL, ...
                        RGTA, RGSD, RGPD, RGOSD, WRGSD, PWAO,
TWAWS, AWI, ...
                        WAD, AMI, WAMF, WAMMD, WAMM, PWS, PWM,
POD, PMD, ...
                        PPD, DWM, DWS, SFS, PDD, PSD, GDDT, GCMF,
GCMMD, ...
                        GCMM, GCWS, PGCO, GCWD, GCMD, CGQI, PGQ,
GCD, ...
                        FAD_F, WAD_F, PPD_F, PSD_F, GCD_F);

WAD_Amount = zeros(1,TS);

syms x

cdf_gevd_WAD(x) = exp(-(1-xiHat_WAD.*((x-
lambdaHat_WAD)./deltaHat_WAD)).^(1./xiHat_WAD));


for i = 1:TS

WAD_Amount(i) = solve(cdf_gevd_WAD == ProbWAD(i),x);

end


% fzero method
% for i = 1:T
%
% fun = @(y_C)(exp(-((1-xiHat_C*((y_C-
lambdaHat_C)/deltaHat_C)).^(1/xiHat_C))) - RCUORiskHi(i));
%
% CEUOR(i) = fzero(fun,y0_C)
%
% end
```

File Name: mle_gevd_product_packaging.m

```
% Generalized extreme value distributions (GEVD) Parameter
Estimate for
% Window Product Packaging Delay

x_gen_PPD = normrnd(mu_PPD, sigma_PPD, n_PPD, 1);

x_gen_PPD2 = zeros(n_PPD,1);

for i = 1:n_PPD

    if x_gen_PPD(i) >= 0;
        x_gen_PPD2(i) = x_gen_PPD(i);
    else x_gen_PPD2(i) = 0;
    end

end

pdf_gevd_PPD = @(x,lambda,delta,xi) ...
    (1./delta).*(exp(-(1-xi.*((x-lambda)./delta)).^(1./xi))).*(1-
xi.*((x-lambda)./delta)).^((1./xi)-1);

start = [lambdaStart_PPD deltaStart_PPD xiStart_PPD];

lb = [];
ub = [];
gevd_paramEsts_PPD = mle(x_gen_PPD, 'pdf',pdf_gevd_PPD,
'start',start, ...
    'lower',lb, 'upper',ub);

lambdaHat_PPD = gevd_paramEsts_PPD(1);
deltaHat_PPD = gevd_paramEsts_PPD(2);
xiHat_PPD = gevd_paramEsts_PPD(3);

% lambdaHat_C = round((gevd_paramEsts(1)*10000))/10000;
% deltaHat_C = round((gevd_paramEsts(1)*10000))/10000;
% xiHat_C = round((gevd_paramEsts(1)*10000))/10000;

ProbPPD = Assembly_Delays(FMSPLR, FMSPA, RFMS, FMOSD, FMPD, FMSD,
FSFS, FSTA, ...
                        WFMSD, FAMMD, MDDT, PFMQ, FMQI, FAMF,
FAMM, TFAWS, ...
                        PFAO, FAWD, FAMD, FAD, RGSPLR, RGSPA,
RGMS, RGSFRTL, ...
                        RGTA, RGSD, RGPD, RGOSD, WRGSD, PWAO,
TWAWS, AWI, ...
                        WAD, AMI, WAMF, WAMMD, WAMM, PWS, PWM,
POD, PMD, ...
                        PPD, DWM, DWS, SFS, PDD, PSD, GDDT, GCMF,
GCMMD, ...
```

```
                        GCMM, GCWS, PGCO, GCWD, GCMD, CGQI, PGQ,
GCD, ...
                        FAD_F, WAD_F, PPD_F, PSD_F, GCD_F);

PPD_Amount = zeros(1,TS);

syms x

cdf_gevd_PPD(x) = exp(-(1-xiHat_PPD.*((x-
lambdaHat_PPD)./deltaHat_PPD)).^(1./xiHat_PPD));


for i = 1:TS

PPD_Amount(i) = solve(cdf_gevd_PPD == ProbPPD(i),x);

end


% fzero method
% for i = 1:T
%
% fun = @(y_C)(exp(-((1-xiHat_C*((y_C-
lambdaHat_C)/deltaHat_C)).^(1/xiHat_C))) - RCUORiskHi(i));
%
% CEUOR(i) = fzero(fun,y0_C)
%
% end
```

File Name: mle_gevd_product_shipping_test.m

```
% Generalized extreme value distributions (GEVD) Parameter
Estimate for
% Window Product Shipping Delay


x_gen_PSD = normrnd(mu_PSD, sigma_PSD, n_PSD, 1);

x_gen_PSD2 = zeros(n_PSD,1);

for i = 1:n_PSD

    if x_gen_PSD(i) >= 0;
        x_gen_PSD2(i) = x_gen_PSD(i);
    else x_gen_PSD2(i) = 0;
    end
```

```
end

pdf_gevd_PSD = @(x,lambda,delta,xi) ...
    (1./delta).*(exp(-(1-xi.*((x-lambda)./delta)).^(1./xi))).*(1-
xi.*((x-lambda)./delta)).^((1./xi)-1);

start = [lambdaStart_PSD deltaStart_PSD xiStart_PSD];

lb = [];
ub = [];
gevd_paramEsts_PSD = mle(x_gen_PSD, 'pdf',pdf_gevd_PSD,
'start',start, ...
    'lower',lb, 'upper',ub);

lambdaHat_PSD = gevd_paramEsts_PSD(1);
deltaHat_PSD = gevd_paramEsts_PSD(2);
xiHat_PSD = gevd_paramEsts_PSD(3);

% lambdaHat_C = round((gevd_paramEsts(1)*10000))/10000;
% deltaHat_C = round((gevd_paramEsts(1)*10000))/10000;
% xiHat_C = round((gevd_paramEsts(1)*10000))/10000;

ProbPSD = Assembly_Delays(FMSPLR, FMSPA, RFMS, FMOSD, FMPD, FMSD,
FSFS, FSTA, ...
                        WFMSD, FAMMD, MDDT, PFMQ, FMQI, FAMF,
FAMM, TFAWS, ...
                        PFAO, FAWD, FAMD, FAD, RGSPLR, RGSPA,
RGMS, RGSFRTL, ...
                        RGTA, RGSD, RGPD, RGOSD, WRGSD, PWAO,
TWAWS, AWI, ...
                        WAD, AMI, WAMF, WAMMD, WAMM, PWS, PWM,
POD, PMD, ...
                        PPD, DWM, DWS, SFS, PDD, PSD, GDDT, GCMF,
GCMMD, ...
                        GCMM, GCWS, PGCO, GCWD, GCMD, CGQI, PGQ,
GCD, ...
                        FAD_F, WAD_F, PPD_F, PSD_F, GCD_F);

PSD_Amount = zeros(1,TS);

syms x

cdf_gevd_PSD(x) = exp(-(1-xiHat_PSD.*((x-
lambdaHat_PSD)./deltaHat_PSD)).^(1./xiHat_PSD));


for i = 1:TS

PSD_Amount(i) = solve(cdf_gevd_PSD == ProbPSD(i),x);

end
```

```
% fzero method
% for i = 1:T
%
% fun = @(y_C)(exp(-((1-xiHat_C*((y_C-
lambdaHat_C)/deltaHat_C)).^(1/xiHat_C))) - RCUORiskHi(i));
%
% CEUOR(i) = fzero(fun,y0_C)
%
% end
```

File Name: Assembly_Equilibrium.m

```
WFMIR = zeros(200,TS); % Window Frame Materials Inflow Rate
WFAR = zeros(200,TS); % Window Frame Assembly Rate
AFCR = zeros(200,TS); % Assembled Frame Consuming Rate
RGIR = zeros(200,TS); % Raw Glass Inflow Rate
WGCR = zeros(200,TS); % Window Glass Cutting Rate
CGCR = zeros(200,TS); % Cutted Glass Consuming Rate
FWPR = zeros(200,TS); % Finished Window Packaging Rate
PWSR = zeros(200,TS); % Packaged Window Shipping Rate
WFMOR = zeros(200,TS); % Window Frame Materials Ordering Rate
RGOR = zeros(200,TS); % Raw Glass Ordering Rate

WFMI = zeros(200,TS); % Window Frame Materials Inventory
RGI = zeros(200,TS); % Raw Glass Inventory
AWFI = zeros(200,TS); % Assembled Window Frame Inventory
CWGI = zeros(200,TS); % Cutted Window Glass Inventory
FWI = zeros(200,TS); % Finished Window Inventory
PWI = zeros(200,TS); % Packaged Window Inventory

q = zeros(1,12); % Optimization Variables
A = zeros(10,12);
b = zeros(10,1);
fval = zeros(200,7,TS);
Opt_Obj_Val = zeros(1,TS);
q_result = zeros(200,12,TS);
% q_hampel = zeros(200,12,TS);
demand = zeros(200,TS);

for i = 1:TS
```

```
    %%%%%%%%%%%%%%%%
    % Optimization %
    %%%%%%%%%%%%%%%%
    %
    %% Cost & Demand Equations
    %
    % f1(q_11) = MRMPC(1)*q_11 + 10
    % f2(q_22) = MRMPC(2)*q_22 + 20
    % c11(q_11) = 0.5*q_11 + 3.5 (don't need anymore)
    % c22(q_22) = 0.5*q_22 + 3.5 (don't need anymore)
    % g1(q_11) = 0.3*q_11 + 1.5
    % g2(q_22) = 0.1*q_22 + 0.5
    % h(q_1k) = 0.5*min(q_1k,q_2k) + 8.5
    % p(q_kl) = 8.5*q_kl + 0.5
    % s(q_lm) = 4*q_lm + 6
    %
        % Market Demand
%         d(i) = demand_0 - price_slope*q(12);

        f1 = MRMPC(1,1)*(q(1)-WFMSD_Amount(i))^2 ...
            + MRMPC(1,2)*(q(1)-WFMSD_Amount(i))*(q(2)-
WRGSD_Amount(i)) ...
            + MRMPC(1,3)*(q(1)-WFMSD_Amount(i));
        f2 = MRMPC(2,1)*(q(2)-WRGSD_Amount(i))^2 ...
            + MRMPC(2,2)*(q(2)-WRGSD_Amount(i))*(q(1)-
WFMSD_Amount(i)) ...
            + MRMPC(2,3)*(q(2)-WRGSD_Amount(i));
        g1 = MWAPC(1,1)*(q(3)-FAD_Amount(i))^2 ...
            + MWAPC(1,2)*(q(3)-FAD_Amount(i)) + MWAPC(1,3);
        g2 = MWAPC(2,1)*(q(4)-GCD_Amount(i))^2 ...
            + MWAPC(2,2)*(q(4)-GCD_Amount(i)) + MWAPC(2,3);
        h = MWAPC(3,1)*min((q(5)-WAD_Amount(i))^2,(q(6)-
WAD_Amount(i))^2) ...
            + MWAPC(3,2)*min((q(5)-WAD_Amount(i)),(q(6)-
WAD_Amount(i))) ...
            + MWAPC(3,3);
        p = MWAPC(4,1)*(q(7)-PPD_Amount(i))^2 ...
            + MWAPC(4,2)*(q(7)-PPD_Amount(i)) + MWAPC(4,3);
        s = MWAPC(5,1)*(q(8)-PSD_Amount(i))^2 ...
            + MWAPC(5,2)*(q(8)-PSD_Amount(i)) + MWAPC(5,3);

    %% Variables:
    %
    % Q = {q_d1,q_d2,q_11,q_22,q_1k,q_2k,q_kl,q_lm}
    % ROU = {rou_11,rou_22,rou_d,rou_m}
    %
    % <=> q = zeros(1,12)
    % q(1) = q_d1 = WFMOR; q(2) = q_d2 = WRGOR; q(3) = q_11; q(4)
= q_22; q(5) = q_1k;
    % q(6) = q_2k; q(7) = q_kl; q(8) = q_lm; q(9) = rou_11; q(10)
= rou_22;
```

```
    % q(11) = rou_d; q(12) = rou_m;
    %% Constraints:
    %   Flow Constraints:
    %
    %   d(rou_m) = demand_0 - price_slope*rou_m <= 100*q_lm
    %   q_d1 <= 4*d(rou_m) = 4*(demand_0 - price_slope*rou_m)
    %   q_d2 <= (1/3)*d(rou_m) = (1/3)*(demand_0 -
price_slope*rou_m)
    %   4*q_11 <= q_1k
    %   q_22 <= 3*q_2k
    %   q_kl <= 10*q_1k
    %   q_kl <= 10*q_2k
    %   q_lm <= 10*q_kl
    %   -q(:,1) <= -WFMSD_Amount(i);
    %   -q(:,2) <= -WRGSD_Amount(i);
    %   -q(:,3) <= -FAD_Amount(i);
    %   -q(:,4) <= -GCD_Amount(i);
    %   -q(:,5) <= -WAD_Amount(i);
    %   -q(:,6) <= -WAD_Amount(i);
    %   -q(:,7) <= -PPD_Amount(i);
    %   -q(:,8) <= -PSD_Amount(i);
    %
    %   Manufacturer's Selling Price
    %
    %   sum(Unit Processing Costs) + sum(Raw Material Costs) <=
rou_m   <=>
    %   sum(MWAPC) + rou_11 + rou_22 <= rou_m
    %
    %
    %   Consumer's Willing to Pay <= Manufacturer's Selling Price
    %
    %   rou_d <= rou_m
    %
        A = [0 0 0 0 0 0 0 100 0 0 0 price_slope;
            1 0 0 0 0 0 0 0 0 0 0 4*price_slope;
            0 1 0 0 0 0 0 0 0 0 0 (1/3)*price_slope;
            0 0 -1 0 4 0 0 0 0 0 0 0;
            0 0 0 -3 0 1 0 0 0 0 0 0;
            0 0 0 0 -10 0 1 0 0 0 0 0;
            0 0 0 0 0 -10 1 0 0 0 0 0;
            0 0 0 0 0 0 -10 1 0 0 0 0;
            0 0 0 0 0 0 0 0 1 1 0 -1;
            0 0 0 0 0 0 0 0 0 0 1 -1;
            -1 0 0 0 0 0 0 0 0 0 0 0;
            0 -1 0 0 0 0 0 0 0 0 0 0;
            0 0 -1 0 0 0 0 0 0 0 0 0;
            0 0 0 -1 0 0 0 0 0 0 0 0;
            0 0 0 0 -1 0 0 0 0 0 0 0;
            0 0 0 0 0 -1 0 0 0 0 0 0;
            0 0 0 0 0 0 -1 0 0 0 0 0;
            0 0 0 0 0 0 0 -1 0 0 0 0;];
```

```
        b = [demand_0 + 100*PSD_Amount(i);
             4*demand_0 + WFMSD_Amount(i);
             (1/3)*demand_0 + WRGSD_Amount(i);
             4*WAD_Amount(i) - FAD_Amount(i);
             WAD_Amount(i) - 3*GCD_Amount(i);
             PPD_Amount(i) - 10*WAD_Amount(i);
             PPD_Amount(i) - 10*WAD_Amount(i);
             PSD_Amount(i) - 10*PPD_Amount(i);
             -sum(MWAPC(:,1));
             0;
             -WFMSD_Amount(i);
             -WRGSD_Amount(i);
             -FAD_Amount(i);
             -GCD_Amount(i);
             -WAD_Amount(i);
             -WAD_Amount(i);
             -PPD_Amount(i);
             -PSD_Amount(i)];

        Aeq = [];
        beq = [];
        lb = zeros(1,12);
        ub = [];

    %% Objective Functions:
    %
    %   Supplier Stage (-Min):
    %     Max (rou_11*q_d1 + rou_22*q_d2) - (f1 + f2)
    %   Raw Material Assembly Stage: Min g1*q_11 + g2*q_22
    %   Windows Assembly Stage: Min h*q_1k
    %   Windows Packaging Stage: Min p*q_kl
    %   Windows Shipping Stage: Min s*q_lm
    %   Manufacturer as a Whole (-Min): Max q_lm*rou_m
    %   Market Stage: Min q_lm*rou_d
    %
        y(1) = -(((q(9)*(q(1)-WFMSD_Amount(i))) + q(10)*(q(2)-
WRGSD_Amount(i))) ...
            - (f1 + f2));
        y(2) = g1*(q(3)-FAD_Amount(i)) + g2*(q(4)-GCD_Amount(i));
        y(3) = h*min((q(5)-WAD_Amount(i)),(q(6)-WAD_Amount(i)));
        y(4) = p*(q(7)-PPD_Amount(i));
        y(5) = s*(q(8)-PSD_Amount(i));
        y(6) = -(q(8)-PSD_Amount(i))*q(12);
        y(7) = (q(8)-PSD_Amount(i))*q(11);

    %% Genetic Algorithm Generating Pareto Front
    FitnessFunction = @(q)[y(1),y(2),y(3),y(4),y(5),y(6),y(7)];
    numberOfVariables = 12;


    % options =
optimoptions(@gamultiobj,'PlotFcn',{@gaplotpareto});
```

```
    [q,fval(:,:,i)] =
gamultiobj(FitnessFunction,numberOfVariables,A,b,Aeq,beq,lb,ub);

%     % Obtain optimization solutions with hampel filter to
remove outliers
%     q_hampel(:,:,i) = hampel(q);

    %%%%%%%%%%%%%%%%%%%
    % System Dynamics %
    %%%%%%%%%%%%%%%%%%%
    %% Flows
    WFMOR(:,i) = q(:,1);
    RGOR(:,i) = q(:,2);
    WFMIR(:,i) = WFMOR(:,i) - WFMSD_Amount(i);
    RGIR(:,i) = RGOR(:,i) - WRGSD_Amount(i);
    WFAR(:,i) = q(:,3) - FAD_Amount(i);
    WGCR(:,i) = q(:,4) - GCD_Amount(i);
    AFCR(:,i) = q(:,5) - WAD_Amount(i);
    CGCR(:,i) = q(:,6) - WAD_Amount(i);
    FWPR(:,i) = q(:,7) - PPD_Amount(i);
    PWSR(:,i) = q(:,8) - PSD_Amount(i);

    %% Inventory Levels
    if i == 1;
        WFMI(:,i) = WFMI_0 + WFMIR(:,i) - 4*WFAR(:,i); % 1
Assembled Frame Consumes 4 Frame Materials
        AWFI(:,i) = AWFI_0 + WFAR(:,i) - AFCR(:,i);
        RGI(:,i) = RGI_0 + RGIR(:,i) - (1/3)*WGCR(:,i); % 1 Raw
Glass Cut into 3 Glass Pieces
        CWGI(:,i) = CWGI_0 + WGCR(:,i) - CGCR(:,i);
        FWI(:,i) = FWI_0 + AFCR(:,i) - 10*FWPR(:,i); % 10
Finished Windows Packaged into 1 Batch
        PWI(:,i) = PWI_0 + FWPR(:,i) - 10*PWSR(:,i); % 10 Batches
of Windows Shipped in 1 Truck
    else
        WFMI(:,i) = WFMI(:,i-1) + WFMIR(:,i) - 4*WFAR(:,i); % 1
Assembled Frame Consumes 4 Frame Materials
        AWFI(:,i) = AWFI(:,i-1) + WFAR(:,i) - AFCR(:,i);
        RGI(:,i) = RGI(:,i-1) + RGIR(:,i) - (1/3)*WGCR(:,i); % 1
Raw Glass Cut into 3 Glass Pieces
        CWGI(:,i) = CWGI(:,i-1) + WGCR(:,i) - CGCR(:,i);
        FWI(:,i) = FWI(:,i-1) + AFCR(:,i) - 10*FWPR(:,i); % 10
Finished Windows Packaged into 1 Batch
        PWI(:,i) = PWI(:,i-1) + FWPR(:,i) - 10*PWSR(:,i); % 10
Batches of Windows Shipped in 1 Truck
    end

    q_result(:,:,i) = q(:,:);

    demand(:,i) = demand_0 - price_slope*q(:,12);
```

```
    Opt_Obj_Val(i) = (-1)*fval(1,1,i)+(-
1)*fval(1,6,i)+fval(1,7,i) ...
        - (fval(1,2,i)+fval(1,3,i)+fval(1,4,i)+fval(1,5,i));

end

% Flows
trans_WFMOR = WFMOR';
trans_RGOR = RGOR';
trans_WFMIR = WFMIR';
trans_RGIR = RGIR';
trans_WFAR = WFAR';
trans_WGCR = WGCR';
trans_AFCR = AFCR';
trans_CGCR = CGCR';
trans_FWPR = FWPR';
trans_PWSR = PWSR';

% Inventory Levels
trans_WFMI = WFMI';
trans_AWFI = AWFI';
trans_RGI = RGI';
trans_CWGI = CWGI';
trans_FWI = FWI';
trans_PWI = PWI';

Inv_Lvls = zeros(12,200,6);
Inv_Lvls(:,:,1) = trans_WFMI;
Inv_Lvls(:,:,2) = trans_AWFI;
Inv_Lvls(:,:,3) = trans_RGI;
Inv_Lvls(:,:,4) = trans_CWGI;
Inv_Lvls(:,:,5) = trans_FWI;
Inv_Lvls(:,:,6) = trans_PWI;
```

File Name: Assembly_Equilibrium_no_equil.m

```
WFMIR = zeros(200,TS); % Window Frame Materials Inflow Rate
WFAR = zeros(200,TS); % Window Frame Assembly Rate
AFCR = zeros(200,TS); % Assembled Frame Consuming Rate
RGIR = zeros(200,TS); % Raw Glass Inflow Rate
WGCR = zeros(200,TS); % Window Glass Cutting Rate
CGCR = zeros(200,TS); % Cutted Glass Consuming Rate
FWPR = zeros(200,TS); % Finished Window Packaging Rate
```

```
PWSR = zeros(200,TS); % Packaged Window Shipping Rate
WFMOR = zeros(200,TS); % Window Frame Materials Ordering Rate
RGOR = zeros(200,TS); % Raw Glass Ordering Rate

WFMI = zeros(200,TS); % Window Frame Materials Inventory
RGI = zeros(200,TS); % Raw Glass Inventory
AWFI = zeros(200,TS); % Assembled Window Frame Inventory
CWGI = zeros(200,TS); % Cutted Window Glass Inventory
FWI = zeros(200,TS); % Finished Window Inventory
PWI = zeros(200,TS); % Packaged Window Inventory

q = zeros(1,12); % Optimization Variables
% A = zeros(10,12);
% b = zeros(10,1);
fval = zeros(200,7,TS);
Opt_Obj_Val_no_equil = zeros(1,TS);
q_result = zeros(200,12,TS);
% q_hampel = zeros(200,12,TS);
demand = zeros(200,TS);

for i = 1:TS

    %%%%%%%%%%%%%%%%
    % Optimization %
    %%%%%%%%%%%%%%%%
    %
    %% Cost & Demand Equations
    %
    % f1(q_11) = MRMPC(1)*q_11 + 10
    % f2(q_22) = MRMPC(2)*q_22 + 20
    % c11(q_11) = 0.5*q_11 + 3.5 (don't need anymore)
    % c22(q_22) = 0.5*q_22 + 3.5 (don't need anymore)
    % g1(q_11) = 0.3*q_11 + 1.5
    % g2(q_22) = 0.1*q_22 + 0.5
    % h(q_1k) = 0.5*min(q_1k,q_2k) + 8.5
    % p(q_kl) = 8.5*q_kl + 0.5
    % s(q_lm) = 4*q_lm + 6
    %
        % Market Demand
%         d(i) = demand_0 - price_slope*q(12);

        f1 = MRMPC(1,1)*(q(1)-WFMSD_Amount(i))^2 ...
            + MRMPC(1,2)*(q(1)-WFMSD_Amount(i))*(q(2)-
WRGSD_Amount(i)) ...
            + MRMPC(1,3)*(q(1)-WFMSD_Amount(i));
        f2 = MRMPC(2,1)*(q(2)-WRGSD_Amount(i))^2 ...
            + MRMPC(2,2)*(q(2)-WRGSD_Amount(i))*(q(1)-
WFMSD_Amount(i)) ...
            + MRMPC(2,3)*(q(2)-WRGSD_Amount(i));
        g1 = MWAPC(1,1)*(q(3)-FAD_Amount(i))^2 ...
            + MWAPC(1,2)*(q(3)-FAD_Amount(i)) + MWAPC(1,3);
        g2 = MWAPC(2,1)*(q(4)-GCD_Amount(i))^2 ...
```

```
            + MWAPC(2,2)*(q(4)-GCD_Amount(i)) + MWAPC(2,3);
        h = MWAPC(3,1)*min((q(5)-WAD_Amount(i))^2,(q(6)-
WAD_Amount(i))^2) ...
            + MWAPC(3,2)*min((q(5)-WAD_Amount(i)),(q(6)-
WAD_Amount(i))) ...
            + MWAPC(3,3);
        p = MWAPC(4,1)*(q(7)-PPD_Amount(i))^2 ...
            + MWAPC(4,2)*(q(7)-PPD_Amount(i)) + MWAPC(4,3);
        s = MWAPC(5,1)*(q(8)-PSD_Amount(i))^2 ...
            + MWAPC(5,2)*(q(8)-PSD_Amount(i)) + MWAPC(5,3);

    %% Variables:
    %
    % Q = {q_d1,q_d2,q_11,q_22,q_1k,q_2k,q_kl,q_lm}
    % ROU = {rou_11,rou_22,rou_d,rou_m}
    %
    % <=> q = zeros(1,12)
    % q(1) = q_d1 = WFMOR; q(2) = q_d2 = WRGOR; q(3) = q_11; q(4)
= q_22; q(5) = q_1k;
    % q(6) = q_2k; q(7) = q_kl; q(8) = q_lm; q(9) = rou_11; q(10)
= rou_22;
    % q(11) = rou_d; q(12) = rou_m;
    %% Constraints:
    %   Flow Constraints:
    %
    %   d(rou_m) = demand_0 - price_slope*rou_m <= 100*q_lm
    %   q_d1 <= 4*d(rou_m) = 4*(demand_0 - price_slope*rou_m)
    %   q_d2 <= (1/3)*d(rou_m) = (1/3)*(demand_0 -
price_slope*rou_m)
    %   4*q_11 <= q_1k
    %   q_22 <= 3*q_2k
    %   q_kl <= 10*q_1k
    %   q_kl <= 10*q_2k
    %   q_lm <= 10*q_kl
    %   -q(:,1) <= -WFMSD_Amount(i);
    %   -q(:,2) <= -WRGSD_Amount(i);
    %   -q(:,3) <= -FAD_Amount(i);
    %   -q(:,4) <= -GCD_Amount(i);
    %   -q(:,5) <= -WAD_Amount(i);
    %   -q(:,6) <= -WAD_Amount(i);
    %   -q(:,7) <= -PPD_Amount(i);
    %   -q(:,8) <= -PSD_Amount(i);
    %
    %   Manufacturer's Selling Price
    %
    %   sum(Unit Processing Costs) + sum(Raw Material Costs) <=
rou_m   <=>
    %   sum(MWAPC) + rou_11 + rou_22 <= rou_m
    %
    %
    %   Consumer's Willing to Pay <= Manufacturer's Selling Price
    %
```

```
%   rou_d <= rou_m
%
    A = [0 0 0 0 0 0 0 100 0 0 0 price_slope;
        1 0 0 0 0 0 0 0 0 0 0 4*price_slope;
        0 1 0 0 0 0 0 0 0 0 0 (1/3)*price_slope;
        0 0 -1 0 4 0 0 0 0 0 0 0;
        0 0 0 -3 0 1 0 0 0 0 0 0;
        0 0 0 0 -10 0 1 0 0 0 0 0;
        0 0 0 0 0 -10 1 0 0 0 0 0;
        0 0 0 0 0 0 -10 1 0 0 0 0;
        -1 0 0 0 0 0 0 0 0 0 0 0;
        0 -1 0 0 0 0 0 0 0 0 0 0;
        0 0 -1 0 0 0 0 0 0 0 0 0;
        0 0 0 -1 0 0 0 0 0 0 0 0;
        0 0 0 0 -1 0 0 0 0 0 0 0;
        0 0 0 0 0 -1 0 0 0 0 0 0;
        0 0 0 0 0 0 -1 0 0 0 0 0;
        0 0 0 0 0 0 0 -1 0 0 0 0;];

    b = [demand_0 + 100*PSD_Amount(i);
         4*demand_0 + WFMSD_Amount(i);
         (1/3)*demand_0 + WRGSD_Amount(i);
         4*WAD_Amount(i) - FAD_Amount(i);
         WAD_Amount(i) - 3*GCD_Amount(i);
         PPD_Amount(i) - 10*WAD_Amount(i);
         PPD_Amount(i) - 10*WAD_Amount(i);
         PSD_Amount(i) - 10*PPD_Amount(i);
         -WFMSD_Amount(i);
         -WRGSD_Amount(i);
         -FAD_Amount(i);
         -GCD_Amount(i);
         -WAD_Amount(i);
         -WAD_Amount(i);
         -PPD_Amount(i);
         -PSD_Amount(i)];

    Aeq = [];
    beq = [];
    lb = zeros(1,12);
    ub = [];

%% Objective Functions:
%
%   Supplier Stage (-Min):
%     Max (rou_11*q_d1 + rou_22*q_d2) - (f1 + f2)
%   Raw Material Assembly Stage: Min g1*q_11 + g2*q_22
%   Windows Assembly Stage: Min h*q_1k
%   Windows Packaging Stage: Min p*q_kl
%   Windows Shipping Stage: Min s*q_lm
%   Manufacturer as a Whole (-Min): Max q_lm*rou_m
%   Market Stage: Min q_lm*rou_d
%
```

```
        y(1) = -(((q(9)*(q(1)-WFMSD_Amount(i))) + q(10)*(q(2)-
WRGSD_Amount(i))) ...
            - (f1 + f2));
        y(2) = g1*(q(3)-FAD_Amount(i)) + g2*(q(4)-GCD_Amount(i));
        y(3) = h*min((q(5)-WAD_Amount(i)),(q(6)-WAD_Amount(i)));
        y(4) = p*(q(7)-PPD_Amount(i));
        y(5) = s*(q(8)-PSD_Amount(i));
        y(6) = -(q(8)-PSD_Amount(i))*q(12);
        y(7) = (q(8)-PSD_Amount(i))*q(11);

    %% Genetic Algorithm Generating Pareto Front
    FitnessFunction = @(q)[y(1),y(2),y(3),y(4),y(5),y(6),y(7)];
    numberOfVariables = 12;


    % options =
optimoptions(@gamultiobj,'PlotFcn',{@gaplotpareto});
    [q,fval(:,:,i)] =
gamultiobj(FitnessFunction,numberOfVariables,A,b,Aeq,beq,lb,ub);

%     % Obtain optimization solutions with hampel filter to
remove outliers
%     q_hampel(:,:,i) = hampel(q);

    %%%%%%%%%%%%%%%%%%%
    % System Dynamics %
    %%%%%%%%%%%%%%%%%%%
    %% Flows
    WFMOR(:,i) = q(:,1);
    RGOR(:,i) = q(:,2);
    WFMIR(:,i) = WFMOR(:,i) - WFMSD_Amount(i);
    RGIR(:,i) = RGOR(:,i) - WRGSD_Amount(i);
    WFAR(:,i) = q(:,3) - FAD_Amount(i);
    WGCR(:,i) = q(:,4) - GCD_Amount(i);
    AFCR(:,i) = q(:,5) - WAD_Amount(i);
    CGCR(:,i) = q(:,6) - WAD_Amount(i);
    FWPR(:,i) = q(:,7) - PPD_Amount(i);
    PWSR(:,i) = q(:,8) - PSD_Amount(i);

    %% Inventory Levels
    if i == 1;
        WFMI(:,i) = WFMI_0 + WFMIR(:,i) - 4*WFAR(:,i); % 1
Assembled Frame Consumes 4 Frame Materials
        AWFI(:,i) = AWFI_0 + WFAR(:,i) - AFCR(:,i);
        RGI(:,i) = RGI_0 + RGIR(:,i) - (1/3)*WGCR(:,i); % 1 Raw
Glass Cut into 3 Glass Pieces
        CWGI(:,i) = CWGI_0 + WGCR(:,i) - CGCR(:,i);
        FWI(:,i) = FWI_0 + AFCR(:,i) - 10*FWPR(:,i); % 10
Finished Windows Packaged into 1 Batch
        PWI(:,i) = PWI_0 + FWPR(:,i) - 10*PWSR(:,i); % 10 Batches
of Windows Shipped in 1 Truck
    else
```

```
        WFMI(:,i) = WFMI(:,i-1) + WFMIR(:,i) - 4*WFAR(:,i); % 1
Assembled Frame Consumes 4 Frame Materials
        AWFI(:,i) = AWFI(:,i-1) + WFAR(:,i) - AFCR(:,i);
        RGI(:,i) = RGI(:,i-1) + RGIR(:,i) - (1/3)*WGCR(:,i); % 1
Raw Glass Cut into 3 Glass Pieces
        CWGI(:,i) = CWGI(:,i-1) + WGCR(:,i) - CGCR(:,i);
        FWI(:,i) = FWI(:,i-1) + AFCR(:,i) - 10*FWPR(:,i); % 10
Finished Windows Packaged into 1 Batch
        PWI(:,i) = PWI(:,i-1) + FWPR(:,i) - 10*PWSR(:,i); % 10
Batches of Windows Shipped in 1 Truck
    end

    q_result(:,:,i) = q(:,:);

    demand(:,i) = demand_0 - price_slope*q(:,12);

    Opt_Obj_Val_no_equil(i) = (-1)*fval(1,1,i)+(-
1)*fval(1,6,i)+fval(1,7,i) ...
        - (fval(1,2,i)+fval(1,3,i)+fval(1,4,i)+fval(1,5,i));

end

% Flows
trans_WFMOR = WFMOR';
trans_RGOR = RGOR';
trans_WFMIR = WFMIR';
trans_RGIR = RGIR';
trans_WFAR = WFAR';
trans_WGCR = WGCR';
trans_AFCR = AFCR';
trans_CGCR = CGCR';
trans_FWPR = FWPR';
trans_PWSR = PWSR';

% Inventory Levels
trans_WFMI = WFMI';
trans_AWFI = AWFI';
trans_RGI = RGI';
trans_CWGI = CWGI';
trans_FWI = FWI';
trans_PWI = PWI';

Inv_Lvls = zeros(12,200,6);
Inv_Lvls(:,:,1) = trans_WFMI;
Inv_Lvls(:,:,2) = trans_AWFI;
Inv_Lvls(:,:,3) = trans_RGI;
Inv_Lvls(:,:,4) = trans_CWGI;
Inv_Lvls(:,:,5) = trans_FWI;
Inv_Lvls(:,:,6) = trans_PWI;
```

Acknowledgment of Previous Publications

All work presented in this dissertation shows the original work of the author. Nevertheless, some concepts of this dissertation have been published or submitted for publications elsewhere. The following presents the works needed to be acknowledged.

Sun, K., & Luxhøj, J. T. (2016). An Integrated Dynamic Uncertainty Approach for Supply Chain Risk Analysis. *The Industrial and Systems Engineering Research Conference.* Anaheim, California: Institute of Industrial and Systems Engineers.

Sun, K., & Luxhøj, J. T. (Submitted). A Hybrid Approach for Modelling Dynamic Flows and Systemic Risks in Supply Chains. *International Journal of Industrial and Systems Engineering*. Submitted February, 2017.

Sun, K., & Luxhøj, J. T. (2017). An Optimization-Enhanced Dynamic Approach for Supply Chain. *Proceedings of the 2017 Industrial and Systems Engineering Conference.* Pittsburgh, PA: Institute of Industrial and Systems Engineers.

Sun, K., & Luxhøj, J. T. Optimizing Dynamic Flows with Systemic Risks in Supply Chains: A Network Equilibrium Approach. Working Paper, *Rutgers University, Department of Industrial and Systems Engineering*.

Bibliography

Abdelgawad, M., & Fayek, A. R. (2011). Comprehensive Hybrid Framework for Risk Analysis in the Construction Industry Using Combined Failure Mode and Effect Analysis, Fault Trees, Event Trees, and Fuzzy Logic. *Journal of Construction Engineering and Management, 138*(5), 642-651.

Ahn, H. S., & Kaminsky, P. (2005). Production and Distribution Policy in a Two-stage Stochastic Push-pull Supply Chain. *IIE Transactions, 37*(7), 609-621.

Alfaro-Cid, E., Castillo, P. A., Esparcia, A., Sharman, K., Merelo, J. J., Prieto, A., & Laredo, J. J. (2008). Comparing Multiobjective Evolutionary Ensembles for Minimizing Type I and II Errors for Bankruptcy Prediction. *2008 IEEE Congress on Evolutionary Computation* (pp. 2902-2908). IEEE.

Allen, D. P. (2009). A frequency domain Hampel filter for blind rejection of sinusoidal interference from electromyograms. *Journal of Neuroscience Methods, 177*(2), 303-310.

Amid, A., Ghodsypour, S. H., & O'Brien, C. (2006). Fuzzy Multiobjective Linear Model for Supplier Selection in a Supply Chain. *International Journal of Production Economics, 104*(2), 394-407.

Anderson, E. G., Fine, C. H., & Parker, G. G. (2000). Upstream volatility in the supply chain: The machine tool industry as a case study. *Production and Operations Management, 9*(3), 239-261.

Andrews, J. D., & Dunnett, S. J. (2000). Event-tree analysis using binary decision diagrams. *IEEE Transactions on Reliability, 49*(2), 230-238.

Angerhofer, B. J., & Angelides, M. C. (2000). System dynamics modelling in supply chain management: research review. *Simulation Conference, 2000* (pp. 342-351). Winter: IEEE.

Antunes, R., & Gonzalez, V. (2015). A Production Model for Construction: A Theoretical Framework. *Buildings, 5*(1), 209-228.

Apostolakis, G. E. (2004). How useful is quantitative risk assessment? *Risk Analysis, 24*(3), 515-520.

Asadi, E., da Silva, M. G., Antunes, C. H., Dias, L., & Glicksman, L. (2014). Multi-objective optimization for building retrofit: A model using genetic algorithm and artificial neural network and an application. *Energy and Buildings, 81*, 444-456.

Baranyi, P., Várkonyi-Kóczy, A. R., Yam, Y., & Patton, R. J. (2005). Adaptation of TS fuzzy models without complexity expansion: HOSVD-based approach. *IEEE Transactions on Instrumentation and Measurement, 54*(1), 52-60.

Barkan, C., Anderson, R., & Tyler, D. (2003). Railroad derailment factors affecting hazardous materials transportation risk. *Transportation Research Record: Journal of the Transportation Research Board, 1825*, 64-74.

Bass, T., & Robichaux, R. (2001). Defense-in-depth revisited: qualitative risk analysis methodology for complex network-centric operations. *Military Communications Conference 2001 Communications for Network-Centric Operations: Creating the Information Force. 1,* pp. 64-70. IEEE.

Beamon, B. M. (1998). Supply chain design and analysis: Models and methods. *International journal of production economics, 55*(3), 284-294.

Bedford, T., & Cooke, R. M. (2001). *Probabilistic Risk Analysis: Foundations and Methods.* Cambridge: Cambridge University Press.

Benaroch, M., Lichtenstein, Y., & Robinson, K. (2006). Real options in information technology risk management: an empirical validation of risk-option relationships. *Mis Quarterly*, 827-864.

Brandenburg, M. (2013). Quantitative models for value-based supply chain management. In *Supply Management Research* (pp. 149-172). Springer Fachmedien Wiesbaden.

Brandenburg, M. (2013). Quantitative models for value-based supply chain management. *Supply Management Research*, 149-172.

Brealey, R. A., Myers, S. C., Allen, F., & Mohanty, P. (2012). *Principles of corporate finance*. Tata McGraw-Hill Education.

Cakravastia, A., & Diawati, L. (1999). Development of system dynamic model to diagnose the logistic chain performance of shipbuilding industry in Indonesia. *In International System Dynamics Conference.* Wellington, New Zealand.

Castillo, E., Hadi, A. S., Balakrishnan, N., & Sarabia, J. M. (2005). *Extreme value and related models with applications in engineering and science.* Hoboken: Wiley.

Čepin, M., & Mavko, B. (2002). A dynamic fault tree. *Reliability Engineering & System Safety, 75*(1), 83-91.

Chakraborty, D., Jana, D. K., & Roy, T. K. (2015). Multi-item integrated supply chain model for deteriorating items with stock dependent demand under fuzzy random and bifuzzy environments. *Computers & Industrial Engineering, 88,* 166-180.

Chang, C. T., Chou, Y. Y., & Zhuang, Z. Y. (2015). A practical expected-value-approach model to assess the relevant procurement costs. *Journal of the Operational Research Society, 66*(4), 539-553.

Chen, C. T., Lin, C. T., & Huang, S. F. (2006). A fuzzy approach for supplier evaluation and selection in supply chain management. *International journal of production economics, 102*(2), 289-301.

Chen, F., Drezner, Z., Ryan, J. K., & Simchi-Levi, D. (2000). Quantifying the bullwhip effect in a simple supply chain: The impact of forecasting, lead times, and information. *Management science, 46*(3), 436-443.

Chen, J. J., & Kodell, R. L. (1989). Quantitative risk assessment for teratological effects. *Journal of the American Statistical Association, 84*(408), 966-971.

Chen, Y. C., & Juang, J. C. (2012). Outlier-Detection-Based Indoor Localization System for Wireless Sensor Networks. *International Journal of Navigation and Observation, 2012*, 11 pages.

Chopra, S., & Meindl, P. (2007). Supply chain management. Strategy, planning & operation. In *Das Summa Summarum des Management* (pp. 265-275). Gabler.

Chopra, S., & Meindl, P. (2010). *Supply Chain Management: Strategy, Planning, and Operation* (4th ed.). Upper Saddle River: Prentice Hall.

Chopra, S., & Sodhi, M. S. (2004). Managing risk to avoid supply-chain breakdown. *MIT Sloan management review, 46*(1), 53.

Christensen, S. S., Agarwal, R., De Carvalho, E., & Cioffi, J. M. (2008). Weighted sum-rate maximization using weighted MMSE for MIMO-BC beamforming design. *IEEE Transactions on Wireless Communications, 7*(12), 4792-4799.

Cleden, M. D. (2012). *Managing project uncertainty*. Gower Publishing, Ltd.

Clemens, P., & Simmons, R. (1998). Lesson IX-Event Tree Analysis. In *System Safety and Risk Management: A Guide for Engineering Educators* (pp. IX-7). Cincinnati: U.S. Department of Health and Human Services.

Cox, L. T., Babayev, D., & Huber, W. (2005). Some limitations of qualitative risk rating systems. *Risk Analysis, 25*(3), 651-662.

Crouhy, M., Galai, D., & Mark, R. (2006). *The Essentials of Risk Management*. New York: McGraw Hill Professional.

Cummings, G. E. (1975). Application of the Fault Tree Technique to A Nuclear Reactor Containment System. In R. E. Barlow, & J. B. Fussell, *Reliability and Fault Tree Analysis* (pp. 805-825). Philadelphia: Siam.

Dagum, P., Galper, A., & Horvitz, E. (1992). Dynamic network models for forecasting. *the eighth international conference on uncertainty in artificial intelligence* (pp. 41-48). Morgan Kaufmann Publishers Inc.

Daultani, Y., Kumar, S., Vaidya, O. S., & Tiwari, M. K. (2015). A supply chain network equilibrium model for operational and opportunism risk mitigation. *International Journal of Production Research, 53*(18), 5685-5715.

De Moraes, R. M., & dos Santos Machado, L. (2004). Using fuzzy hidden Markov models for online training evaluation and classification in virtual reality simulators. *International Journal of General Systems, 33*(2-3), 281-288.

Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. Berlin: Springer.

Dixon, P., & Ellison, A. M. (1996). Introduction: ecological applications of Bayesian inference. *Ecological Applications, 6*(4), 1034-1035.

Dong, J., Zhang, D., & Nagurney, A. (2004). A supply chain network equilibrium model with random demands. *European Journal of Operational Research, 156*(1), 194-212.

Edwards, W. (1954). The theory of decision making. *Psychological bulletin, 51*(4), 380.

Eierdanz, F., Alcamo, J., Acosta-Michlik, L., Krömker, D., & Tänzler, D. (2008). Using fuzzy set theory to address the uncertainty of susceptibility to drought. *Regional Environmental Change, 8*(4), 197-205.

Featherston, C. R., & Doolan, M. (2012). A Critical Review of the Criticisms of System Dynamics. *The 30th International Conference of the System Dynamics Society.* St Gallen, Switzerland: System Dynamics Society.

Ferdous, R., Khan, F., Sadiq, R., Amyotte, P., & Veitch, B. (2011). Fault and event tree analyses for process systems risk analysis: uncertainty handling formulations. *Risk Analysis, 31*(1), 86-107.

Fonseca, C. M., & Fleming, P. J. (1993). Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Genelralization. In S. Forrest (Ed.), *Genetic Algorithms: Proceedings of the Fifth International Conference.* San Mateo, CA: Morgan Kaufmann.

Forrester, J. W. (1997). Industrial dynamics. *Journal of the Operational Research Society, 48*(10), 1037-1041.

Fung, R., & Chang, K. C. (2013). *Weighing and integrating evidence for stochastic simulation in Bayesian networks.* Ithaca, NY: arXiv. Retrieved from arXiv.org.

Furuta, H., & Shiraishi, N. (1984). Fuzzy Importance in Fault Tree Analysis. *Fuzzy Sets and Systems, 12*, 205-213.

Fussell, J. B. (1976). Fault tree analysis: concepts and techniques. In H. H. Waite, *Pressure vessels and piping: design and analysis* (p. VI). United States: American Society of Mechanical Engineers.

Gazdik, I. (1985). Fault Diagnosis and Prevention by Fuzzy Sets. *IEEE Trans. on Reliability, 34*, 382-388.

Gencay, R., & Selcuk, F. (2004). Extreme value theory and Value-at-Risk: Relative performance in emerging markets. *International Journal of Forecasting, 20*(2), 287-303.

Georgiadis, P., Vlachos, D., & Iakovou, E. (2005). A system dynamics modeling framework for the strategic supply chain management of food chains. *Journal of Food Engineering, 70*(3), 351-364.

Gilli, M., & Këllezi, E. (2006). An application of extreme value theory for measuring financial risk. *Computational Economics, 27*(2-3), 207-228.

Goodman, G. R. (1988). An Assessment of Coal Mine Escapeway Reliability Using Fault Tree Analysis. *Mining Science and Technology, 7*(2), 205-215.

Gorsevski, P. V., Gessler, P. E., & Jankowski, P. (2003). Integrating a fuzzy k-means classification and a Bayesian approach for spatial prediction of landslide hazard. *Journal of geographical systems, 5*(3), 223-251.

Guillén, G., Mele, F. D., Bagajewicz, M. J., Espuna, A., & Puigjaner, L. (2005). Multiobjective supply chain design under uncertainty. *Chemical Engineering Science, 60*(6), 1535-1553.

Hadi, A. S., Balakrishnan, N., & Sarabia, J. M. (2005). *Extreme value and related models with applications in engineering and science*. Hoboken: Wiley.

Hahn, G. J., & Kuhn, H. (2012). Value-based performance and risk management in supply chains: A robust optimization approach. *International Journal of Production Economics, 139*(1), 135-144.

Hamdouch, Y. (2011). Multi-period supply chain network equilibrium with capacity constraints and purchasing strategies. *Transportation Research Part C: Emerging Technologies, 19*(5), 803-820.

Hammond, D., & Beullens, P. (2007). Closed-loop supply chain network equilibrium under legislation. *European journal of operational research, 183*(2), 895-908.

Hampel, F. R. (1974). The influence curve and its role in robust estimation. *Journal of the American Statistical Association, 69*(346), 383-393.

Han, Z. Y., & Weng, W. G. (2010). An integrated quantitative risk analysis method for natural gas pipeline network. *Journal of Loss Prevention in the Process Industries, 23*(3), 428-436.

Harkleroad, E., Vela, A., Kuchar, J., Barnett, B., & Merchant-Bennett, R. (2013). *Risk-based Modeling to Support NextGen Concept Assessment and Validation.* Lexington, Massachusetts: Lincoln Lab Report, ATC-405.

Hassan, M. R. (2009). A combination of hidden Markov model and fuzzy model for stock market forecasting. *Neurocomputing, 72*(16), 3439-3446.

Hausman, W. H. (2002). Supply Chain Performance Metrics. In C. Billington, T. Harrison, H. Lee, & J. Neale (Eds.), *The Practice of Supply Chain Management* (pp. 61-73). Verlag: Springer US.

Ho, W., Zheng, T., Yildiz, H., & Talluri, S. (2015). Supply chain risk management: a literature review. *International Journal of Production Research, 53*(16), 5031-5069.

Holland, J. H. (1975). *Adaptation in natural and artificial systems.* Ann Arbor, MI: University of Michigan Press.

Horn, J., Nafpliotis, N., & Goldberg, D. E. (1994). A Niched Pareto Genetic Algorithm for Multiobjective Optimization. Piscataway, NJ: IEEE.

Hosking, J. M., Wallis, J. R., & Wood, E. F. (1985). Estimation of the generalized extreme-value distribution by the method of probability-weighted moments. *Technometrics, 27*(3), 251-261.

Hudoklin, A., & Rozman, V. (1992). Safety analysis of the railway traffic system. *Reliability Engineering and System Safety, 37*(1), 7-13.

Hui, K. T., & Liu, D. B. (2004). A Bayesian belief network model and tool to evaluate risk and impact in software development projects. *Reliability and maintainability, 2004 annual symposium-RAMS* (pp. 297-301). IEEE.

Joyce, P., Rokyta, D. R., Beisel, C. J., & Orr, H. A. (2008). A general extreme value theory model for the adaptation of DNA sequences under strong selection and weak mutation. *Genetics, 180*(3), 1627-1643.

Jüttner, U. (2005). Supply chain risk management: Understanding the business requirements from a practitioner perspective. *The International Journal of Logistics Management, 16*(1), 120-141.

Jüttner, U., Peck, H., & Christopher, M. (2003). Supply Chain Risk Management: Outlining an Agenda for Future Research. *International Journal of Logistics: Research and Applications, 6*, 197-210.

Kahn, S. U. (2009). A Self-adaptive Weighted Sum Technique for the Joint Optimization of Performance and Power Consumption in Data Centers. *ISCA PDCCS*, (pp. 13-18).

Kaki, A., Salo, A., & Talluri, S. (2015). Disruptions in Supply Networks: A Probabilistic Risk Assessment. *Journal of Business Logistics, 36*(3), 273-287.

Kangwanpongpan, T., da Silva, R. C., & Krautz, H. J. (2012). Prediction of oxy-coal combustion through an optimized weighted sum of gray gases model. *Energy, 41*(1), 244-251.

Kaplan, S., & Garrick, B. J. (1981). On the quantitative definition of risk. *Risk Analysis, 1*, 11-28.

Keynes, J. M. (1936). *The General Theory of Employment, Interest and Money*. London: Macmillan and Co.

Khan, F. (2009). *LTE for 4G mobile broadband: air interface technologies and performance.* Cambridge, UK: Cambridge University Press.

Khan, F. I., & Abbasi, S. A. (2000). Analytical simulation and PROFAT II: a new methodology and a computer automated tool for fault tree analysis in chemical process industries. *Journal of Hazardous Materials, 75*(1), 1-27.

Kim, I. Y., & De Weck, O. L. (2005). Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. *Structural and multidisciplinary optimization, 29*(2), 149-158.

Kjaerulff, U. B., & Madsen, A. L. (2008). *Bayesian networks and influence diagrams.* New York: Springer Science+ Business Media.

Klugman, S. A., Panjer, H. H., & Willmot, G. E. (2012). *Loss models: from data to decisions* (Vol. 715). John Wiley & Sons.

Kolar, C. S., & Lodge, D. M. (2002). Ecological predictions and risk assessment for alien fishes in North America. *Science, 298*(5596), 1233-1236.

Konak, A., Coit, D. W., & Smith, A. E. (2006). Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety, 91*(9), 992-1007.

Korpelevich, G. M. (1977). The extragradient method for finding saddle points and other problems. *Matekon, 13*, 35-49.

Kumar, M., Vrat, P., & Shankar, R. (2006). A fuzzy programming approach for vendor selection problem in a supply chain. *International Journal of Production Economics, 101*(2), 273-285.

La Cour, B. R. (2004). Statistical characterization of active sonar reverberation using extreme value theory. *IEEE Journal of Oceanic Engineering, 29*(2), 310-316.

Lederer, P. J., & Li, L. (1997). Pricing, production, scheduling, and delivery-time competition. *Operations Research, 45*(3), 407-420.

Lee, C. C. (1990). Fuzzy logic in control systems: fuzzy logic controller. II. *IEEE Transactions on Systems, Man and Cybernetics*, *20*(2), 419-435.

Lee, E., Park, Y., & Shin, J. G. (2009). Large engineering project risk management using a Bayesian belief network. *Expert Systems with Applications, 36*(3), 5880-5887.

Lee, H. L., & Billington, C. (1993). Material management in decentralized supply chains. *Operations Research, 41*(5), 835-847.

Lin, C. T., Chiu, H., & Chu, P. Y. (2006). Agility index in the supply chain. *International Journal of Production Economics, 100*(2), 285-299.

Lindhe, A., Rosen, L., Norberg, T., & Bergstedt, O. (2009). Fault tree analysis for integrated and probabilistic risk analysis of drinking water systems. *Water Research*, 1641-1653.

Liu, H., Shah, S., & Jiang, W. (2004). On-line outlier detection and data cleaning. *Computers & chemical engineering, 28*(9), 1635-1647.

Lorenz, E. N. (1963). Deterministic nonperiodic flow. *Journal of the atmospheric sciences*, *20*(2), 130-141.

Lorenz, E. N. (1972). Predictability; Does the Flap of a Butterfly's wings in Brazil Set off a Tornado in Texas? *American Association for the advancement of science 139th meeting.*

Luxhøj, J. T. (2005). Model-Based Reasoning for Aviation Safety Risk Assessments. *SAE 2005 Transactions Journal of Aerospace*.

Luxhøj, J. T., & Sarlo, R. (2012). Decision Analysis for Safety Risk Modeling of Unmanned Aircraft Systems. *Proceedings of the Industrial and Systems Engineering Research Conference.* Orlando, FL.

Luxhøj, J. T., Jalil, M., & Jones, S. M. (2003). A risk-based decision support tool for evaluating aviation technology integration in the national airspace system. *Proceedings of the AIAA's 3rd Annual Aviation Technology, Integration, and Operations (ATIO) Technical Forum* (pp. 17-19). Denver, CO: AIAA.

Maiti, M. K., & Maiti, M. (2006). Fuzzy inventory model with two warehouses under possibility constraints. *Fuzzy Sets and Systems, 157*(1), 52-73.

Mamdani, E. H. (1977). Application of fuzzy logic to approximate reasoning using linguistic synthesis. *IEEE Transactions on Computers, 100*(12), 1182-1191.

Marcot, B. G., Steventon, J. D., Sutherland, G. D., & McCann, R. K. (2006). Guidelines for developing and updating Bayesian belief networks applied to ecological modeling and conservation. *Canadian Journal of Forest Research, 36*(12), 3063-3074.

Marimoutou, V., Raggad, B., & Trabelsi, A. (2009). Extreme value theory and value at risk: application to oil market. *Energy Economics, 31*(4), 519-530.

Marler, R. T., & Arora, J. S. (2010). The weighted sum method for multi-objective optimization: new insights. *Structural and multidisciplinary optimization, 46*(1), 853-862.

Mass, N. (1980). Stock and flow variables and the dynamics of supply and demand. In J. Randers, *Elements of the System Dynamics Method.* Waltham: Pegasus Communications.

McBratney, A. B., & Odeh, I. O. (1997). Application of fuzzy sets in soil science: fuzzy logic, fuzzy measurements and fuzzy decisions. *Geoderma*, *77*(2), 85-113.

McCann, R. K., Bruce, G. M., & Rick, E. (2006). Bayesian Belief Networks: Applications in Ecology and Natural Resource Management. *Canadian Journal of Forest Research, 36*(12), 3053-3062.

McDonald, K. S., Ryder, D. S., & Tighe, M. (2015). Developing best-practice Bayesian Belief Networks in ecological risk assessments for freshwater and estuarine ecosystems: A quantitative review. *Journal of environmental management, 154*, 190-200.

McNab, B., & Alves, D. (2016, March 20). *Risk Assessment Frameworks—Food Safety Risk Assessment*. Retrieved from Government of Ontario: http://www.gov.on.ca/OMAFRA/english/research/risk/frameworks/as2c.html#

McNeil, A. J., & Frey, R. (2000). Estimation of tail-related risk measures for heteroscedastic financial time series: an extreme value approach. *Journal of empirical finance, 7*(3), 271-300.

Mendel, J. M. (1995). Fuzzy logic systems for engineering: a tutorial. *IEEE*, (pp. 345-377).

Mum, J. (2006). *Modeling risk: Applying Monte Carlo simulation, real options analysis, forecasting, and optimization techniques* (Vol. 347). John Wiley & Sons.

Murata, T., Ishibuchi, H., & Tanaka, H. (1996). Multi-objective genetic algorithm and its applications to flowshop scheduling. *Computers & Industrial Engineering, 30*(4), 957-968.

Nagurney, A., & Toyasaki, F. (2005). Reverse supply chain management and electronic waste recycling: a multitiered network equilibrium framework for e-cycling. *Transportation Research Part E: Logistics and Transportation Review*, *41*(1), 1-28.

Nagurney, A., Dong, J., & Zhang, D. (2002). A supply chain network equilibrium model. *Transportation Research Part E: Logistics and Transportation Review, 38*(5), 281-303.

Nagurney, A., Yu, M., Folden, J., & Nagurney, L. S. (2014). Supply chain network competition in time-sensitive markets. *Transportation Research Part E: Logistics and Transportation Review, 70*, 112-127.

Naidu, K., Mokhlis, H., & Bakar, A. A. (2014). Multiobjective optimization using weighted sum artificial bee colony algorithm for load frequency control. *International Journal of Electrical Power & Energy Systems, 55*, 657-667.

Nash, J. F. (1950). Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences, 36*, pp. 48-49. USA.

Nielsen, T. D., & Jensen, F. V. (2009). *Bayesian networks and decision graphs.* Springer Science & Business Media.

Olson, D. L., & Wu, D. D. (2010). A Review of Enterprise Risk Management in Supply Chain. *Kybernetes, 39*, 694-706.

Oztekin, A. E., & Luxhøj, J. T. (2010). An inductive reasoning approach for building system safety risk models of aviation accidents. *Journal of Risk Research, 13*(4), 479-499.

Pan, H., & Yun, W. (1997). Fault tree analysis with fuzzy gates. *Computers & industrial engineering, 33*(3), 569-572.

Pappis, C. P., & Mamdani, E. H. (1977). A fuzzy logic controller for a trafc junction. *IEEE Transactions on Systems, Man and Cybernetics, 7*(10), 707-717.

Pearl, J. (1986). Fusion, propagation, and structuring in belief networks. *Artificial Intelligence, 29*(3), 241-288.

Pearson, R. K. (2002). Outliers in process modeling and identification. *IEEE Transactions on control systems technology, 10*(1), 55-63.

Poston, R., & Severin, G. (2001). Financial impacts of enterprise resource planning implementations. *International Journal of Accounting Information Systems, 2*(4), 271-294.

Puigbò, P., Bravo, I. G., & Garcia-Vallvé, S. (2008). E-CAI: a novel server to estimate an expected value of Codon Adaptation Index (eCAI). *BMC bioinformatics, 9*(1), 1.

Ramzali, N., Lavasani, M. M., & Ghodousi, J. (2015). Safety barriers analysis of offshore drilling system by employing Fuzzy Event Tree Analysis. *Safety Science, 78*, 49-59.

Rangaiah, G. P. (2009). *Multi-objective Optimization: Techniques And Applications In Chemical Engineering.* Hackensack: World Scientific.

Rao, M. C., & Prahlad, V. (1997). A tunable fuzzy logic controller for vehicle-active suspension. *Fuzzy sets and systems, 85*(1), 11-21.

Ravindran, A. R., & Warsing Jr, D. P. (2012). *Supply chain engineering: Models and applications.* CRC Press.

Richardson, G. (1991). *Feedback Thought in Social Science and Systems Theory*. Philadelphia: University of Pennsylvania Press.

Ritchie, B., & Brindley, C. (2007). Supply chain risk management and performance: A guiding framework for future development. *International Journal of Operations & Production Management, 27*(3), 303-322.

Rochet, J. C. (2003). Why Are There So Many Banking Crises? *CESifo Economic Studies, 49*(2), 141-155.

Ross, T. J. (2009). *Fuzzy logic with engineering applications.* John Wiley & Sons.

Sahay, N. (2016). *Hybrid Simulation Based Optimization for Supply Chain Management.* Rutgers University. Rutgers University Library.

Schilling, G. (1996). Working capital's role in maintaining corporate liquidity. *TMA Journal, 16*(5), 4-7.

Shoshanah, C., & Roussel, J. (2005). *Strategic Supply Chain Management: The Five Disciplines For Top Performance.* New York.

Simon, H. A. (1959). Theories of decision-making in economics and behavioral science. *The American economic review*, *49*(3), 253-283.

Singer, A. (2010). *External report reviewing the previous opinions of the Panel on Animal Health and Welfare concerning the application of quantitative tools, in the sequence of the current self mandate on "good practice in conducting scientific assessments in animal health.* EFSA European Food Safety Authority.

Singhvi, A., & Shenoy, U. V. (2002). Aggregate planning in supply chains by pinch analysis. *Chemical Engineering Research and Design*, *80*(6), 597-605.

Smith, K. V. (1973). State of the art of working capital management. *Financial management, 2*(3), 50-55.

Snyman, J. (2005). *Practical mathematical optimization: an introduction to basic optimization theory and classical and new gradient-based algorithms* (Vol. 97). Springer Science & Business Media.

Sterman, J. D. (2000). *Business dynamics: systems thinking and modeling for a complex world.* Boston: Irwin/McGraw-Hill.

Steuer, R. E., & Na, P. (2003). Multiple criteria decision making combined with finance: A categorized bibliographic study. *European Journal of operational research, 150*(3), 496-515.

Summer, M. (2000). Risk factors in enterprise-wide/ERP projects. *Journal of information technology*, *15*(4), 317-327.

Sun, K., & Luxhøj, J. T. (2016). An Integrated Dynamic Uncertainty Approach for Supply Chain Risk Analysis. *The Industrial and Systems Engineering Research Conference.* Anaheim, California: Institute of Industrial and Systems Engineers.

Sun, K., & Luxhøj, J. T. (Submitted). A Hybrid Approach for Modelling Dynamic Flows and Systemic Risks in Supply Chains. *International Journal of Industrial and Systems Engineering*. Submitted February, 2017.

Sun, K., & Luxhøj, J. T. (2017). An Optimization-Enhanced Dynamic Approach for Supply Chain. *Proceedings of the 2017 Industrial and Systems Engineering Conference.* Pittsburgh, PA: Institute of Industrial and Systems Engineers.

Tanaka, H., Fan, L. T., & Taguchi, K. (1983). Fault Tree Analysis by Fuzzy Probability. *IEEE Trans. on Reliability, 32*, 453-457.

Tang, Y., & Xu, L. (1994). Fuzzy logic application for intelligent control of a variable speed drive. *IEEE Transactions on Energy Conversion*, *9*(4), 679-685.

Teng, C. X., Yao, F. M., & Hu, X. W. (2007). Study on multi-commodity flow supply chain network equilibrium model with random demand. *Systems Engineering-Theory & Practice, 27*(10), 77-83.

Thun, J., & Hoenig, D. (2011). An Empirical Analysis of Supply Chain Risk Management in the German Automotive Industry. *International Journal of Production Economics, 131*, 242-249.

Ticehurst, J. L., Newham, L. T., Rissik, D., Letcher, R. A., & Jakeman, A. J. (2007). A Bayesian Network Approach for Assessing the Sustainability of Coastal Lakes in New South Wales, Australia. *Environmental Modelling & Software, 22*(8), 1129-1139.

Tichehurst, J. L., Newham, L. T., Rissik, D., Letcher, R. A., & Jakeman, A. J. (2007). A Bayesian network approach for assessing the sustainability of coastal lakes in New South Wales, Australia. *Environmental Modelling & Software*, *22*(8), 1129-1139.

Torres, O. C., & Morán, F. (2006). *The bullwhip effect in supply chains: A review of methods, components and cases.* Palgrave Macmillan.

Towler, E., Rajagopalan, B., Gilleland, E., Summers, R. S., Yates, D., & Katz, R. W. (2010). Modeling hydrologic and water quality extremes in a changing climate: A statistical approach based on extreme value theory. *Water Resources Research*, *46*(11).

Trkman, P., & McCormack, K. (2009). Supply Chain Risk in Turbulent Environments – A Conceptual Model for Managing Supply Chain Network Risk. *International Journal of Production Economics, 119*, 247-258.

Trucco, P., Cagno, E., Ruggeri, F., & Grande, O. (2008). A Bayesian Belief Network modelling of organisational factors in risk analysis: A case study in maritime transportation. *Reliability Engineering & System Safety, 93*(6), 845-856.

Tsujimura, Y., & Gen, M. (1994). Fuzzy Fault Tree and Its Importance Analysis. *18th ICC & IE*, (pp. 301-304).

Uusitalo, L. (2007). Advantages and challenges of Bayesian networks in environmental modelling. *Ecological modelling*, *203*(3), 312-318.

Vesely, W., Goldberg, F., Roberts, N., & Nasel, D. (1981). *Fault Tree Handbook: System and Reliability Research Office of Nuclear Regulatory Research.* US Nuclear Regulatory Commission.

Vick, S. G. (2002). *Degrees of belief: Subjective probability and engineering judgment.* ASCE Publications.

Vos, B. (1997). Redesigning international manfacturing and logistics structures. *International Journal of Physical Distribution & Locistics Management*, *27*(7), 377-394.

Vos, B., & Akkermans, H. A. (1996). Capturing the dynamics of facility allocation. *International Journal of Operations & Production Management*, *16*(11), 57-70.

Wang, T., & Vandendorpe, L. (2011). Iterative resource allocation for maximizing weighted sum min-rate in downlink cellular OFDMA systems. *IEEE Transactions on Signal Processing, 59*(1), 223-234.

Wang, Y. M., Liu, J., & Elhag, T. M. (2008). An integrated AHP–DEA methodology for bridge risk assessment. *Computers and industrial engineering, 54*(3), 513-525.

White, S. F., Tyler, P. M., Erway, A. K., Botkin, M. L., Kolli, V., Meffert, H., & Blair, J. R. (2016). Dysfunctional representation of expected value is associated with reinforcement-based decision-making deficits in adolescents with conduct problems. *Journal of Child Psychology and Psychiatry*.

Whitney, D. E., Luo, J., & Heller, D. A. (2014). The benefits and constraints of temporary sourcing diversification in supply chain disruption and recovery. *Journal of Purchasing and Supply Management*, *20*(4), 238-250.

Wieland, B., Dhollander, S., Salman, M., & Koenen, F. (2011). Qualitative risk assessment in a datascarce environment: a model to assess the impact of control measures on spread of African Swine. *Preventive Veterinary Medicine, 99*(1), 4-14.

Winkler, H., Kaluza, B., & Schemitsch, H. B. (2006). *Managing economical and ecological goal conflicts: demonstrated in the context of sustainable supply chain networks.* Institut für Wirtschaftswissenschaften, Abteilung Produktions-, Logistik-und Umweltmanagement. Klagenfurt: Universität Klagenfurt.

Wolfson, L. J., Kadane, J. B., & Small, M. J. (1996). Bayesian Environmental Policy Decisions: Two Case Studies. *Ecological Applications, 6*(4), 1056-1066.

Wu, D., & Olson, D. L. (2008). Supply chain risk, simulation, and vendor selection. *International journal of production economics, 114*(2), 646-655.

Wu, T., Blackhurst, J., & Chidambaram, V. (2006). A Model for Inbound Supply Risk Analysis. *Computers in Industry*, *57*, 350-365.

Xu, J., & Zhao, L. (2008). A class of fuzzy rough expected value multi-objective decision making model and its application to inventory problems. *Computers & Mathematics with Applications, 56*(8), 2107-2119.

Yang, T. (2007). *Multi Objective Optimization Models for Managing Supply Risk in Supply Chains.* The Pennsylvania State University. Pennsylvania State University.

Zadeh, L. (1963). Optimality and non-scalar-valued performance criteria. *IEEE transactions on Automatic Control, 8*(1), 59-60.

Zadeh, L. A. (1965). Fuzzy Sets. *Inform. and Control*, *8*, 338-353.

Zalk, D. M., Paik, S. Y., & Swuste, P. (2009). Evaluating the control banding nanotool: a qualitative risk assessment method for controlling nanoparticle exposures. *Journal of Nanoparticle Research, 11*(7), 1685-1704.

Zhang L., Xin, Y., & Liang, Y. C. (2009). Weighted sum rate optimization for cognitive radio MIMO broadcast channels. *IEEE Transactions on Wireless Communications, 8*(6), 2950-2959.

Zheng, Y., Cui, W., & Yang, J. (2015). Optimal reinsurance under distortion risk measures and expected value premium principle for reinsurer. *Journal of Systems Science and Complexity, 28*(1), 122-143.

Zhu, D. L., Yang, H., Li, C. M., & Wang, X. L. (2014). Properties of the multiclass traffic network equilibria under a tradable credit scheme. *Transportation Science, 49*(3), 519-534.