The Pennsylvania State University

The Graduate School

Department of Industrial and Manufacturing Engineering

NETWORK RECONFIGURATION FOR SUPPLY CHAIN RISK MITIGATION

A Dissertation in

Industrial Engineering and Operations Research

by

Satama Sirivunnabood

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

December 2010

The dissertation of Satama Sirivunnabood was reviewed and approved* by the following:

Soundar R.T. Kumara
Allen E. Pearce & Allen M. Pearce Professor of Industrial Engineering
Professor of Computer Science and Engineering
Dissertation Advisor
Chair of Committee

David A. Nembhard
Associate Professor of Industrial and Manufacturing Engineering

Sandeep Purao
Associate Professor of Information Sciences and Technology

Tao Yao
Assistant Professor of Industrial and Manufacturing Engineering

Paul Griffin
Professor of Industrial and Manufacturing Engineering
Peter & Angela Dal Pezzo Department Head Chair

*Signatures are on file in the Graduate School

ABSTRACT

Risks and uncertainties around the globe have been evident as severe threats against reliable operations of supply chain networks. In order to survive the volatile conditions that characterize the current global environment, firms must prepare strategies that mitigate the effects of risks on their supply chain networks. This research proposes supply chain risk mitigation strategy based on supply chain network design under uncertainty and integrated advanced Information Technology (IT) architecture. The main objectives of this research include (1) understanding the impacts of risks on a supply chain network, (2) developing an approach to designing a robust supply chain network under risk, and (3) designing an information system architecture for a supply chain network under risk. In particular, we propose an architecture for network reconfiguration in a supply chain network operating under risk. Such a reconfigurable supply chain network has the ability to respond to risks as they arise and adjust its configuration accordingly to avoid catastrophic failure. To achieve a reconfigurable supply chain, three main components have been developed including an uncertain parameter database in which knowledge about the effects of risks is stored, a stochastic network optimization module that is responsible for determining long-term supply chain network configuration using a two-stage stochastic programming model, and a network configuration controller used to facilitate operations among supply chain nodes. Specifically, the information system of this reconfigurable supply chain network is based on the principle of Service-Oriented Architecture (SOA), such that heterogeneous nodes with diverse enterprise software platforms can communicate and integrate with each other seamlessly.

The first section focuses on obtaining knowledge about the effects of risk on a supply chain. This section proposes a Multi-Agent-base Simulation (MAS) model based on the well-known Gaia methodology and Unified Modeling Language (UML) standard and details it use in simulation experiments to gather knowledge about the network design parameters of the supply chain. In the second part, the knowledge obtained from the MAS model is used to formulate a two-stage stochastic programming model to redesign a supply chain network under risk in the stochastic network optimization module. The Sample Average Approximation (SAA) and L-Shaped decomposition methods are used to solve this supply chain network redesign problem efficiently. Additionally, three different sampling techniques are used to generate random numbers to the SAA method including Simple Random Sampling (SRS), Latin Hypercube Sampling (LHS), and Sobol' sequence technique. Results from numerical experiments showed that the LHS technique are the most efficient for solving small problems, whereas, Sobol' sequence technique is more efficient for addressing moderate or large problems. In the third section, the network configuration obtained from the stochastic network optimization module is implemented on a reconfigurable supply chain network developed based on the SOA principle. The results from simulation experiments show that the network reconfiguration framework via the use of the SOA enables discoverability and interoperability in the supply chain and is capable of improving the supply chain performance in the presence of risk.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I am thankful to my advisor, Dr. Soundar Kumara, whose encouragement, guidance, and support from the beginning to the final stages have enabled me to bring this study to completion. I appreciate too the invaluable advice and critiques that I have received from the dissertation committee, Dr. David Nembhard, Dr. Sandeep Purao, and Dr. Tao Yao, as my research progressed. I am also pleased to thank to Dr. Panida Jirutitijaroen from the National University of Singapore, and Dr. Wasu Glankwamdee from the Singapore–MIT Alliance (SMA) for their guidance in my work on stochastic programming modeling and solutions. My gratitude is also to Machigar Ongtang for her discussion regarding the design and deployment of the Service-Oriented Architecture (SOA). The many insights of my colleagues at Penn State all contributed to this study, particularly colleagues at the Department of Industrial and Manufacturing Engineering's Laboratory of Intelligence Systems and Quality (LISQ) and the Department of Anthropology's Richtsmeier Lab. Last of all, I heartily thank my parents, Sonchai and Puntip, and my sisters, Punchada and Pitchaya, who have always encouraged and supported me during my far-from-home study at Penn State.

Chapter 1

Introduction and Research Motivation

Effective supply chain management is an important key to success in today's competitive global markets. Therefore, researchers in both academia and industry have expended much effort in developing approaches and techniques that can be used to design and control supply chain networks effectively. A number of approaches have been developed in recent years with the purpose of improving the performance of supply chains including mathematical optimization, simulation models, and artificial intelligence techniques. However, most of these approaches aim to optimize deterministic supply chains which may not be sufficient in the unpredictable and dynamic conditions of today's global marketplaces.

As supply chain networks today are becoming more global and complex in structure, there is a higher possibility than before that unpredictable events will occur somewhere in a supply chain network (Foroughi et al., 2006). Such unexpected events may cause deviation in the operational conditions or changes in the physical infrastructure of the supply chain network that result in the degradation or catastrophic failure of the supply chain. We refer to these kinds of events, which include, for example, natural disasters, terrorist attacks, fluctuating currency exchange rates, or seasonal customer demand—as *risks in supply chains*.

In the literature, risk and uncertainty in supply chains are sometimes used interchangeably, for example, Sodhi (2005) used the term "risk" to refer to uncertain customer demand, Nagurney et al. (2005) used the term "risk" to represent supply and demand uncertainties, and Chopra et al. (2007) used the terms "risk" and "uncertainty" in similar meaning to explain recurrent and disruption risks/uncertainties. Nevertheless, some researchers have been trying to distinguish their definitions in different ways. It is the fact that there is no consensus on the definitions of risk and uncertainty in supply chains to date. The first approach to risk definition is the one that defines risk as a consequence of uncertain operations or processes in supply chains. Tang (2006) defined supply chain risks into two types: operational risks and disruption risks. The operational risks were referred to the inherent uncertainties such as uncertain demand, uncertain supply, and uncertain cost, whereas, the disruption risks are referred to the major disruptions caused by natural and man-made disasters such as earthquakes, floods, hurricanes, terrorist attacks, etc. Hallikas et al. (2004) stated that risks in supplier networks are the consequence from uncertainty. This uncertainty is mainly from two sources: customer demand and customer deliveries. In other words, they suggested that uncertain demand quantity, delivery cost, delivery time, and product quality will result in the risks in business networks. Norrman and Jansson (2004) referred that "business risk is the level of exposure to uncertainties that the enterprise must understand and effectively manage as it executes its strategies to achieve its business objective and create value", as well as that "risk is the chance, in quantitative terms, of a defined hazard occurring. It therefore combines a probabilistic measure of the occurrence of the primary event(s) with a measure of the consequence of that/those event(s)." Mathematically, they defined that Risk = Probability (of the event)*Business Impact of the event. As a result, while risk can be calculated, uncertainties are genuinely unknown. Gaonkar and Viswanadham (2007) defined supply chain risk using the distribution of the loss resulting from the variation of possible supply chain outcomes, their likelihood, and their subjective values. On the other hand, as another approach to risk definition, Chopra and Sodhi (2004), Kleindorfer and Saad (2005), and Foroughi et al. (2006) referred to risks as events that may occur and cause negative impact on supply chains such as natural disaster, terrorism, strikes, and equipment malfunctions. In particular, these risk events terminate desirable deterministic conditions and make difficulty in managing supply chains.

Due to the variation of the risk and uncertainty definitions discussed above, the concrete definitions of these two terms are given to align readers' perspectives of supply chain risks to ours as we progress through the rest of this research. In our work, we define supply chain risks as events that may possibly occur inside or surrounding a supply chain network and cause negative impact on the supply chain. In particular, the term "risk" and "risk event" will be used interchangeably in the research. These events can be, for example, natural disasters, terrorist attacks, oil crisis, and seasonal change in demand or supply. Further, we argue that once any of these events occurs, it may result in uncertain operations in the supply chain such as uncertain demand, uncertain supply quantity, and uncertain shipping cost. By using this approach in defining the risk, a supply chain network subject to the risk event can be characterized and designed based on such uncertain operations in the supply chain as being explained in the next chapters.

Risk in supply chains is now an ongoing research topic in the supply chain area. Most supply chain practitioners are more interested in establishing and maintaining network reliability rather than network optimality. This is due to the fact that catastrophic failures resulting from the aforementioned events and processes result in much more serious damage than the losses caused by suboptimality in supply chain operations. Such damages not only result in productivity loss on the part of a local working unit, but they also propagate throughout the entire network and hence cause major profitability loss throughout the entire supply chain. Therefore, it is crucial that in order to mitigate risks in the supply chain network they must be taken into account as a primary consideration at early stages of network design.

Supply chains today are more vulnerable than those of the past. This vulnerability is mainly caused by recent trends in supply chain management including Just in Time (JIT) and Lean supply chains, competitive markets, more sophisticated customer demand, outsourcing policies, and global supply chains (Foroughi et al., 2006). The supporting reasons for this argument are as follows:

- JIT and Lean supply chain practices focus on reducing costs and delays in supply chains. For example, let us consider a manufacturing firm implementing both JIT and Lean supply chains. Basically, JIT policy tries to minimize inventories of parts and products and the Lean supply chain aims to develop single-supplier relationships in order to reduce any redundant transactional costs. However, in the presence of risk, this supply chain network would be considered highly vulnerable since it maintains only the required number of parts and products and only one source furnishing supplies. Thus, if the supplier is subject to risk and proves unable to supply raw materials to the manufacturer in the right quantity at the specified time, it is certain that the manufacturer will be unable to continue production and hence will have no choice but to shut down operations for a period of time.
- Furthermore, competitive markets and sophisticated demand also result in higher vulnerability in supply chain networks. A competitive market forces firms to offer more attractive products and services to their customers, for instance, lower prices, on-time delivery, more customized specification, or higher durability—products and services that mean firms incur additional costs. If a firm cannot fulfill these requirements, it may lose the customers. However, if a firm invests too much in order to meet such requirements, it may not make sufficient profit from its sale of goods and/or services. Therefore, the supply chain in this competitive environment is clearly more vulnerable than those of the past.
- Outsourcing is another trend that makes the supply chain vulnerable. To increase their competitiveness, firms are focusing more on their core products and services, such that many firms are outsourcing support tasks or support operations to concentrate on core

competencies. However, outsourcing policy creates a higher degree of dependency in a supply chain network. Therefore, a firm that is engaged in outsourcing will depend more on its suppliers and service providers. This trend consequently imposes risks on the supply chain. For example, a third-party logistics provider may not have an appropriate maintenance schedule for its trucks and/or the provider may not have sufficient trucks to cover its delivery schedule. Given that we can reasonably expect one or more trucks to break down at any given time, it is highly likely that many deliveries will not be made on time with a concomitant negative effect on customer satisfaction and ultimately compromising the possibility of repeat business for the outsourcing firm.

- The last supply chain trend likely to impose risk on supply chains is the trend of global supply chains. Since the material and information in the global supply chains are transferred across different countries, additional possible risks are likely to occur including currency exchange rate, customer preferences, or natural disasters. As a result, the supply chain that operates globally should be prepared for such events and processes that could occur anywhere in the network.

In short, if firms need to improve their supply chain using approaches such as JIT/Lean supply chain, outsourcing, and/or global supply chains, they must also consider the risks to which their networks are subject. One of the most well-known examples of a supply chain risk event is that of the fire that took place at the Royal Philips Electronics' plant in 2000 (Norrman and Jansson, 2004). This plant manufactured microchips and supplied them to two cell phone manufacturers, i.e., Telefon AB L.M. Ericsson and the Nokia Corporation. This fire caused damage to the microchips at the Philips plant and thus rendered the plant unable to supply parts to its customers. As a result, Ericsson temporarily stopped its operations, thereby incurring a loss of about $400 million as indicated in its annual report. On the other hand, Nokia's multiple-supplier policy meant the company suffered very little from this fire crisis: it promptly switched its chip orders from Philips to alternative suppliers. Certainly, Nokia had better risk management strategy which allows them successful in the mobile phone market. Other examples of risk management in practical supply chains include the 1995 earthquake in Kobe, Japan, which resulted in major production losses for several enterprises worldwide, including computer, consumer products, and automobile manufacturers (Sheffi, 2005); the 2004 influenza vaccine shortage in the US (Desroches et al., 2005); and Hewlett-Packard's portfolio approach to reducing risk in procurement (Billington, 2002). In the most recent times during April 2010 entire Europe came to a stand still due to the volcano eruption in Iceland and the resulting six day shutdown of the European airspace. The loss here accounted to from first estimates, 6 Billon dollars.

Chopra and Sodhi (2004) have discussed some effective risk mitigation strategies, for instance, reserving more inventories, having redundant suppliers, and pooling customer demands. These mitigation strategies can be executed either through the action of an individual firm or through coordination among firms in a supply chain network. In this research, the coordination of firms in a supply chain will be emphasized. Specifically, we will focus on the use of supply chain network reconfiguration as an efficient approach to mitigating the supply chain risks. This network reconfiguration approach creates a supply chain network that can withstand and adapt to the risks inherent in operating in an unpredictable and dynamic environment. Examples of network reconfiguration include being able to switch to a different supplier of raw material when the current supplier is subject to a natural disaster, or being able to relocate operations if employees are expected to strike at the current location. The main objectives of this research are shown in Figure 1-1 and an explanation follows.

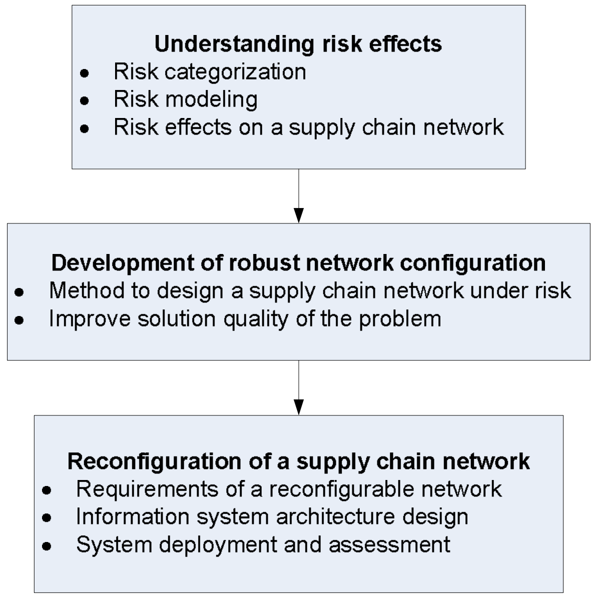

Figure 1-1: Research objectives

1) To understand the effects of risks on a supply chain network: This includes categorizing risks, modeling risks, and also observing the behaviors of a supply chain network once a risk event has occurred. In this research, a Multi-Agent-based Simulation (MAS) model has been developed to study the effects of risks on a supply chain network.

2) To develop a robust network configuration: Once the effects of risks are known, this knowledge will be used to redesign a supply chain network in the presence of risk. The objectives in this regard are to determine a method for redesigning a robust supply chain configuration under risk and also to improve solution quality and computational time for this problem. This study formulates the problem of supply chain network redesign under risk using a two-stage stochastic programming model and solves the problem using a combination of several techniques.

3) To develop a reconfigurable supply chain prototype: Because the network configuration obtained from (2) will change from time to time depending on the type of risks present in the environment, the supply chain network should be able to reconfigure itself in order to withstand new risk. In practice, such a reconfiguration may not be possible due to the

diverse information system platforms used among companies; therefore, in this research, we propose a reconfigurable supply chain architecture that addresses this issue. To design such an architecture, we first define the requirements of a reconfigurable supply chain network. Then, an information system architecture for the reconfigurable supply chain is designed based the principles of Service-Oriented Architecture (SOA). Finally, system deployment and assessment are carried out using simulation experiments.

This research, therefore, focuses on making the following contributions to the study of risk management:

- An improved understanding of the effects of risks on supply chain networks
- An efficient approach to redesigning supply chain networks under risk
- A prototype of an adaptive and self-reconfigurable supply chain network that will mitigate risks

Chapter 2

Problem Description

This research focuses on problems inhering in supply chain network design and implementation at strategic and tactical levels. The strategic level includes decision making related to facility locations and their capacities, supplier selection, and markets served by the supply chain. The tactical level is responsible for determining production planning, order allocation to suppliers, and production allocation to customers. In this chapter, we first discuss a supply chain network operating in a deterministic environment and then identify problems that could arise when the environment becomes uncertain due to the occurrence of a risk event. The research focuses are concluded in the second part of this chapter.

2.1 A deterministic supply chain network

This section considers a problem scenario in which a typical supply chain network with four layers—supplier, plant, warehouse, and customer—delivers a single type of product by transferring it from suppliers to plants, warehouses, and customers consecutively. In particular, no linkage or transportation channel intervenes between the non-immediate layers. The product sold in this supply chain is considered as a commodity and is a low value product (or a common product) such that a buying node can easily procure raw material or product from any available selling node due to the commonality of the product; in addition, the commodity's low per unit cost means that trading negotiations can be conducted online using business enterprise applications without human intervention being necessary. Hence, no significant loss of profit occurs when daily purchasing is not conducted in an optimal way. Figure 2-1 shows an example of a supply chain network under consideration that will be referred to many times throughout this research.

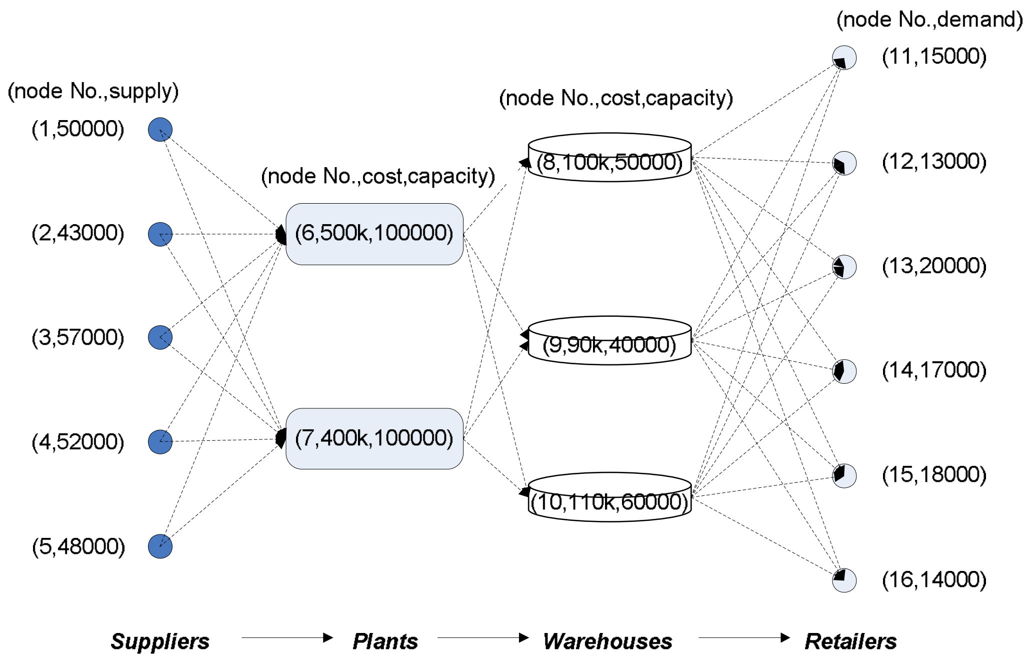

Figure 2-1: An example of a supply chain network

Figure 2-1 provides associated data for each node in the network. In the supplier layer, the node number and supply availability are given for each node. In the plant and warehouse layers, the node number, fixed cost for acquiring a facility at a node, and node capacity are given for each node. Lastly, in the customer layer, each node is associated with its number and customer demand. The objective here is to design a supply chain network that results in minimum facility and shipping costs. Therefore, deterministically, we can formulate this problem as a Mixed Integer Programming (MIP) model and solve it by determining the locations of plants and warehouses, and also supplier and customer nodes, which we procure raw material from and sell product to. To minimize the total facility and shipping costs, we solve the following MIP problem:

Indices and sets:

i, j = index of nodes in the supply chain

S = set of candidate suppliers

P = set of candidate plants

W = set of candidate warehouses

C = set of customers

$F = P \cup W$, i.e. F is a set of all facility nodes

$N_d = S \cup F \cup C$, i.e. N_d is a set of all nodes in the supply chain

A = set of arcs between nodes

Model parameters:

c_i = cost of acquiring a facility at node i

q_{ij} = unit shipping cost of product flowing from node i to node j

d_j = demand at node j

s_i = available supply at node i

r_j = per-unit capacity required when a product is processed at plant or warehouse j

m_j = capacity of facility j

Decision variables:

y_i = 1 if we acquire a facility at node i, otherwise 0. Y = vector of y_i where $i \in F$

x_{ij} = units of product flowing from node i to node j. X = vector of x_{ij} where $i, j \in N_d$

$$\textbf{Min} \ \sum_{i\in F} \boldsymbol{c_i}\, \boldsymbol{y_i} + \sum_{(ij)\in A} \boldsymbol{q_{ij}}\, \boldsymbol{x_{ij}} \tag{2-1}$$

$$s.t \qquad \sum_{i\in N_d} x_{ij} - \sum_{l\in N_d} x_{jl} = 0 \ ; \qquad \forall j \in F \tag{2-2}$$

$$\sum_{i\in W} x_{ij} \ \geq \ d_j \ ; \qquad \forall j \in C \tag{2-3}$$

$$\sum_{j\in P} x_{ij} \ \leq \ s_i \ ; \qquad \forall i \in S \tag{2-4}$$

$$r_j\left(\sum_{i\in N_d} x_{ij}\right) \ \leq \ m_j\, y_j \ ; \qquad \forall j \in F \tag{2-5}$$

$$Y \ \in \{0,1\}^{|F|} \tag{2-6}$$

$$X \in \ R_+^{|A|} \tag{2-7}$$

The objective function of this MIP model is to minimize the sum of the total fixed facility cost and shipping cost as shown in Equation (2-1). Equation (2-2) refers to flow conservation constraints at all facility nodes. Equation (2-3) states that demand at every customer node must be fulfilled. Equation (2-4) states that the units of raw material shipped from each supplier node must not exceed supply availability. A capacity constraint for each facility node is given in Equation (2-5). Equations (2-6) and (2-7) represent binary variables and non-negative variables for facility location and shipping decision variables.

Given the data in Figure 2-1 and the shipping costs shown in Table 2-1, the optimal supply chain network configuration and its shipping routes are shown in Figure 2-2.

Table 2-1: Unit shipping costs in the supply chain

	node j										
node i	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
1	5	4									
2	3	4									
3	3	2									
4	1	4									
5	3	2									
6			7	5	6						
7			5	5	5						
8						10	9	8	9	8	7
9						10	10	9	8	7	6
10						9	9	10	8	7	7

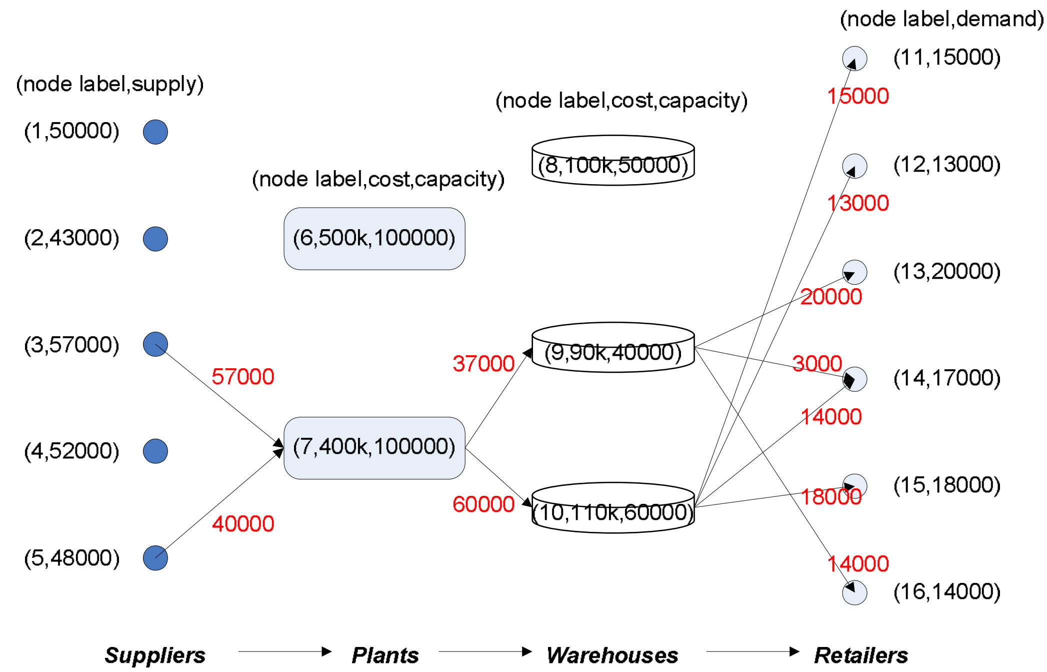

Figure 2-2: Optimal deterministic supply chain network

Typically, the supply chain network shown in Figure 2-2 should operate at a satisfactory level of performance based on the total supply chain facility and operating costs in a deterministic environment. However, if a risk event occurs at one of the nodes in the supply chain, this network configuration may no longer be appropriate. For example, an employee strike at supplier 3 in the network would render the shipping cost and supply availability at supplier 3 uncertain, and hence plant 7 might not be able to procure raw material at its desired price and quantity. Therefore, we need to incorporate the effects of risk into our network design model such that the effects of a risk event on the supply chain are mitigated as well. For the purposes of this research, we assume that only one risk event can occur at a time.

2.2 Research problems

The research problems are divided into 3 parts corresponding to the research objectives in Chapter 1.

2.2.1 Problems in understanding risk effects

In the first part of this research, we study how to categorize supply chain risks such that mitigation strategies can be prepared for each type of risk. An example would be to categorize the risks into disruption risks and operational risks. Though disruption risks are rarely realized (low frequency), when an event associated with them does take place they have a serious impact on supply chain networks. Conversely, operational risks are often realized (high frequency), but they have only a minor impact on supply chains. It is proved in the literature that we need to manage these types of risks differently to optimize the profit of a firm (Chopra, 2007). In this study, therefore, we determine how the risks should be categorized in supply chains. The second problem is how to model the risks in such a dynamic and complex supply chain in order to investigate the supply chain behavior that is subject to risk. Lastly, we observe how each type of risk event affects supply chain behavior. As a result, this knowledge can be used to design a supply chain network that is robust enough to withstand the occurrence of a risk event. As you will see in Chapter 5, to obtain the answers for these questions, a software agent-based supply chain simulation model has been developed and simulation experiments were conducted to observe the behavior of a supply chain.

2.2.2 Problems in developing a robust network configuration

Basically, a supply chain network can be developed that can tolerate risks by two means: reserving more inventory and using appropriate network design methods. In the second section of this research, we will focus on risk-mitigation strategy using a supply chain network design approach. After obtaining the knowledge of risk effects on a supply chain network, we integrate these effects into our supply chain network design model. In this section, too, we propose a method for supply chain redesign in the event of occurrence of a risk. In addition, techniques that can be used to solve the problem efficiently and enhance the solution quality are also studied.

2.2.3 Problems in reconfiguration of a supply chain network

Since each network design or configuration is suitable for each particular type of risk, it is necessary that our supply chain needs to be able to reconfigure or adjust its configuration as risk types change. Each time a network reconfigures itself, new nodes may be included in the network or existing nodes may be removed from the network. Therefore, firms are faced with the problem of heterogeneity and scalability among the information infrastructures of its supply chain partners. In order to establish a network that is reconfigurable, we need an enterprise information architecture that can build the capabilities of interoperability, discoverability, and autonomy into the network. We, therefore, identify the requirements of a reconfigurable supply chain network. And, further, we propose an Information System (IS) architecture for a reconfigurable supply chain network. One of the emerging IS architectures that could be applied to a reconfigurable supply chain network is Service-Oriented Architecture (SOA)—an architecture that has the properties of interoperability, loose coupling, discoverability, autonomy, and other characteristics necessary for the kind of network reconfiguration ability this research seeks to establish.

Therefore, our proposed IS architecture is primarily based on the principles of SOA. Lastly, we demonstrate how to deploy this reconfigurable supply chain network in practice and also assess the benefits obtained from this architecture using simulation experiments.

The details of how we addressed these research problems are given in Chapters 5, 6, and 7.

Chapter 3

Literature Review

Two main research areas in the literature can be considered related to the development of a reconfigurable supply chain network, i.e., supply chain risk management and Service-Oriented Architecture (SOA) in supply chains. In this chapter, reviews of supply chain risk management and SOA in supply chains are given in the first and second parts respectively.

Supply chain risk management has been of interest to the supply chain community for approximately a decade, and the literature on the subject covers both qualitative and quantitative approaches to supply chain problems. In the following sections, we discuss some related studies in this area, including risk management processes in supply chains, risk categorization, risk mitigation strategies, supply chain network design models under risk, and industrial best practices in supply chain risk management.

3.1 Risk management processes in supply chains

Typically, a process of supply chain risk management consists of the following steps:

- Risk identification
- Risk classification
- Risk quantification and assessment
- Risk mitigation

In the literature, each step in a risk management process has been studied using both qualitative and quantitative approaches. For example, Hallikas et al. (2004) proposed a risk management process for a supplier network. Although the main objective of that study was to manage risks in a supplier network, the process can easily be extended to the management of risks in a whole supply chain. The steps of supply chain risk management in Hallikas et al. (2004) include risk identification, risk assessment, decision and implementation of risk management actions, and risk monitoring. The first three steps are self-explanatory and very common steps in risk management processes. However, the last step, risk monitoring, is less frequently referenced: it allows firms to monitor dynamic environments around them. In a dynamic environment, even though a risk has been resolved, it may reoccur, or a new risk may be introduced to the supply chain. Therefore, risk monitoring should be considered as an important step in the supply chain risk management process.

A number of other research studies have focused on the development of supply chain risk management processes. These include a study of the risk management processes of global supply chains by Wang and Yang (2007). A survey cited herein showed that 34% of supply chain risks are internal (logistics, capital, and information), 15% relate to the supplier, 13% relate to the customer, 4% relate to natural events or are governmental, and 6% are related to other sources. In addition, Wang and Yang's study (2007) proposed the risk assessment method. The basic idea of risk assessment is that of calculating the reliability of each node in a supply chain (R_i, where R_i = reliability of node i), and then grouping those nodes according to their responsibilities in the supply chain, e.g., suppliers, manufacturers, or distribution centers. After that, the reliability of each group is calculated using $R_{P_j} = 1 - (R_1 \times R_2 \times \ldots \times R_n)$, where R_{Pj} is the reliability of node group j and n = the total number of nodes in group j. Then, from the reliability of each group, the reliability of the whole network is computed by $R = 1 - (R_{P_1} \times R_{P_2} \times \ldots \times R_{P_n})$, where R is the reliability of the whole network and n = the total number of groups in the whole network. The final solution is the reliability of the whole supply chain, and the complement (L = 1 - R) is the level of risk in the supply chain network considered. Gaonkar and Viswanadham (2004) discussed an analytical framework for supply chain risk management in which supply chain risks were classified into risks from deviations, disruptions, and disasters. Deviations are variations in demand, supply, production, and logistics. Disruptions are events such as earthquakes, hurricanes, and epidemics that cause an infrastructure of a supply chain to change physically. Disasters are defined as temporary irrecoverable shut-downs of the supply chain network due to a catastrophic system-wide disruption such as that resulting from a terrorist attack. Another similar risk management process was proposed by Han and Chen (2007), according to whom the process consists of five components: abnormality and risk identification, risk decomposition and analysis, risk assessment, criteria for optimum risk control, and risk prevention. In addition, this study also proposed a framework for prioritizing risks in supply chains. Further, Stephan and Badr (2007) used the Supply Chain Operational Reference (SCOR) model to measure the total supply chain performance developed by the Supply-Chain Council to extended and integrate risk management into its strategy. The topics integrated into the SCOR model include risk identification, risk reduction, risk contingency planning, and risk monitoring. These areas are very similar to the steps in the risk management processes discussed earlier in this chapter. Kleindorfer and Saad (2005) presented a practical study of supply chain disruption management that identified a process for disruption risk management consisting of (1) specifying risk sources and vulnerabilities, (2) assessing the risks, and (3) mitigating the risks. This study also included an experiment conducted on the U.S. chemical industry during the period of 1995–2000. Based on the results from the empirical data, the study also discussed the design of management systems intended to cope with supply chain disruption risks.

As we can see in this section, a number of studies have been conducted in the area of supply chain risk management processes. However, they are very similar in their approaches and conclusions; that is, they demonstrate that risk management processes in supply chains should include risk identification, classification, quantification and assessment, and mitigation steps as already stated at the beginning of this section.

3.2 Risk categorization in supply chains

In the literature, categorization of risks can be viewed from two perspectives, i.e. from descriptive and modeling purposes. In terms of descriptive purposes, the objective is to classify the risks so that people can understand and identify the events or situations that are potential risks to supply chains. For example, Chopra and Sodhi (2004) classified supply chain risks into nine categories that commonly appear in a supply chain network. These categories include disruptions, delay, systems, forecast, intellectual property, procurement, receivable, inventory, and capacity risks. By closely observing each risk category, a supply chain manager can realize the potential risks in a supply chain network such that it is possible to efficiently prepare mitigation strategies in the face of those risks. For instance, by observing the geographical location of each node in the supply chain network, the manager can estimate the likelihood that natural disasters (disruption risk) will impact the supply chain. Next, the manager can prepare the mitigation strategy for those potential disasters in order to reduce the impact of a disaster to an acceptable level. The other studies that proposed a descriptive categorization of supply chain risks include Tang's (2006) review, wherein risks are categorized as inhering in supply management, demand management, product management, and information management processes. For each type of risk, the author also reviewed many managerial techniques for mitigating the impacts of such risks. Further, Bogataj and Bogataj (2007) classified supply chain risks into five categories: supply risk, process (production and distribution) risk, demand risk, control (quality and time control) risk, and environmental (physical, social, political, legal, operational, economic, and cognitive environments) risk. As we can see, the main objective of the descriptive categorization of supply chain risks is to allow supply chain personnel to understand and subsequently identify the potential risks inhering in supply chains. Such categorization is very beneficial to supply chain managers and practitioners alike. Therefore, managers and practitioners can effectively apply the concepts to improve the robustness of their supply chains. However, in academic research, we still need a more quantitative approach to categorizing supply chain risks to gain more insight into their impacts on supply chain networks. We refer to this approach as categorization for modeling purposes.

Categorization for modeling purposes usually divides supply chain risks based on their frequency with which they are actually realized and the impact their realization has on a supply chain; that is, there are two categories: risks that are realized frequently but have minor impact, and risks that are rarely realized but have serious impact. Basically, we can refer to the former as operational risks, and the latter as disruption risks. In the literature, many researchers categorize supply chain risks in just such a way. For example, in order to prepare robust strategies to mitigate supply chain risks, Tang (2006) divided risks into disruption risks and operational risks. The main distinction in modeling these two risks is that, for disruptions, we should also include jump processes when modeling supply chain parameters, whereas for operational risks, jump processes are not required. He also argued that most of the quantitative models of supply chain risks are designed for managing operational risks and very few address disruption risks. This is due to the fact that (1) a large number of data points have been collected for operational risks, such that probability fitting and further analyses are possible, while on the other hand, very few data points are available for disruption risks; and (2) since disruption risks rarely or may not occur, it may not be worth investing in mitigation programs, after all "Nobody gets credit for fixing problems that never happen." Gaonkar and Viswanadham (2007) categorized supply chain risks into three broad forms: deviation (e.g. demand or supply variation), disruption (e.g., earthquakes and epidemics), and disaster (e.g., terrorist actions). In the present study, we model these risks in a similar way as previously stated. Specifically, the deviation can be modeled as a frequent event with minor impact, and disruption and disaster can be modeled as rare events with major impacts on a supply chain. The main difference between deviation and disruption risks is that disruptions change the physical infrastructure of a supply chain, whereas deviations change some operating parameters to a limited degree. A significant work that illustrated the benefits of classifying supply chain risks into operational and disruption risks is that of Chopra et al. (2007). The authors proposed that supply risks should be considered separately as recurrent (operational) and disruption risks. The quantitative models were set in a problem with one reliable (free from recurrent and disruption risks) but more expensive supplier, one unreliable (subject to both recurrent and disruption risks) supplier, one manufacturer, and fixed demand. Based on these conditions, the study proved that in some situations bundling these two foundational kinds of risks results in the firm refraining from doing business with the reliable supplier, whereas decoupling the two risks results in the firm electing to do business with the reliable supplier. The authors introduced this inequality:

$$\frac{C_o - h}{C_o + e} \leq \frac{h}{C_u - e} \tag{3-1}$$

where C_o = the unit cost of oversupply, C_u = the unit cost of unmet demand, h = the unit cost for reserving raw material, and e = the unit cost for purchasing reserved raw material. If Equation (3-1) holds, bundling the two risks results in the reliable supplier not being used, whereas decoupling the two risks results in the reliable supplier being used. Furthermore, given this realistic problem scenario and assumptions, the authors showed that:

- Bundling the two types of risks (recurrent and disruption) results in the reliable (more expensive) supplier being underutilized and the unreliable (cheaper) one being over utilized.

- As the probability of disruption grows, we should mitigate the risks by purchasing more material from the reliable supplier and less from the cheaper supplier.

Sometimes, researchers categorized risks according to frequency and duration. In fact, we can consider duration as an indicator of the degree of negative impact. That is, a risk event that takes place over a long period of time can be considered as having a serious potential impact, and risk event that takes place over a short time period of time could be treated as having only a minor impact. Tomlin (2006) categorized supply chain risks using frequency and duration and showed that the optimal strategy for risk mitigation depends on the nature of the risk (frequent but short versus rare but long). Based on the nature of the risk, the appropriate mitigation strategies can be provided.

As will be discussed in the later sections, this research emphasizes the categorization of risks for modeling purposes. However, other than the frequency and duration of risk, the location of risk will be also included in the proposed model because, in our work, anywhere in a supply chain network can be subject to risk and different locations impact a supply chain network differently.

3.3 Strategies to mitigate supply chain risks

Once managers realize the effects that given risk events could have on their supply chains, the next step is to develop appropriate mitigation strategies. Chopra and Sodhi (2004) provided a matrix illustrating the relationship between mitigation strategies and each category of risk, as shown in Table 3-1.

Table 3-1: Relationship matrix for mitigation strategies and risks[1]

Mitigation Strategy	Disruption	Delay	Forecast	Procurement	Receivable	Capacity	Inventory
Add capacity		DD		D		II	D
Add inventory	D	DD		D		D	II
Have redundant suppliers	DD			D		I	D
Increase responsiveness		DD	DD				DD
Increase flexibility		D		D		DD	D
Pool demand			DD			DD	DD
Increase capacity		D					D
Have more customers					D		

As we can see in Table 3-1, several risk mitigation strategies are available to reduce the effects of risk events on a supply chain network. However, in the literature, there are two major mitigation strategies for supply chain risks, one is inventory management and another is a network design approach. The following paragraphs give examples of the works related to these two mitigation approaches.

In the inventory management approach, several actions could be performed to mitigate the risks to supply chains. For example, reserving more inventory, determining the order period, and determining order quantity. The basic idea is to create an inventory control system for both parts and final products so that the firm can absorb negative impacts from uncertainty in the supply chains. Özekici and Parlar (1999) developed infinite-horizon periodic-review inventory models with unreliable suppliers where demand, supply, and cost parameters vary because they are dependent on the random changes that take place in an uncertain environment. In their work, the authors considered a single-product inventory system. The state of the environment (I_n) was assumed to follow a time-homogenous Markov chain process. Also, the demand (D_n) was modulated by the Markov chain (I). The main objective was to determine the optimal inventory control policies for a supply chain without ordering cost and with fixed ordering cost. The study

[1] where: D = decreases risk, DD = greatly decreases risk, I = increases risk, and II = greatly increases risk.

showed that in an inventory system with no ordering cost, the so-called base-stock policy leads to the optimality to the expected total discounted cost. That is, there exists $\{S_i\}$ such that

$$y^*(i,x) = \begin{cases} S_i - x, & if\ x \leq S_i \\ 0, & if\ x > S_i \end{cases} \tag{3-2}$$

where y*(i, x) = the optimal ordering policy, i = the current state of the environment, x = the current inventory level, and S_i = the base-stock level. Therefore, for the system with no ordering cost, the expected total discounted cost can be minimized by periodically ordering the parts up to level S_i. In the system with fixed ordering costs, it was shown that the $\{(s_i, S_i)\}$ inventory policy will lead to the optimal expected total discounted cost. That is, there exist s_i and S_i such that

$$y^*(i,x) = \begin{cases} S_i - x, & if\ x \leq s_i \\ 0, & if\ x > s_i \end{cases} \tag{3-3}$$

where s_i = the lower bound of the inventory, and the other notations are the same as the system with no ordering cost. Therefore, for the system with a fixed ordering cost, the expected total discounted cost can be minimized by periodically ordering the parts up to level S_i given that the current inventory is below level s_i. Bertsimas and Thiele (2006) proposed another approach for inventory control in the presence of demand uncertainty, that of authors formulating a robust optimization problem to solve an inventory control problem under demand uncertainty without assuming a specific distribution. The problems under consideration included the cases of single station where the inventory at only one location was considered and a network case where the inventories were considered at several nodes in the supply chain network. The resulting models demonstrated advantages over a traditional dynamic programming method, and they were used to solve the types of problems as stated previously. Jung et al. (2004) developed a simulation-based optimization framework to determine the optimal stock level for each product type in a supply chain of chemical products. The main objective was to set safety stock levels such that a desired level of customer satisfaction could be achieved when the supply chain is subject to demand uncertainty. The major issue here was that of computational complexity. Therefore, simulation-based optimization was selected as the tool for solving this problem. The authors first formulated a multi-stage stochastic programming model to capture the characteristics of the supply chain considered and then decomposed the problem into two parts, i.e. an outer optimization sub-problem and a deterministic planning model. Demand uncertainty was represented by a number of demand scenarios sampled from the Monte Carlo technique. The deterministic planning model was responsible for solving production plans, job scheduling, and customer satisfaction levels for each demand scenario. The outer optimization module then found the optimal customer satisfaction level and safety stock level accordingly. Discrete-event system simulation was used to implement production and scheduling plans from the deterministic planning model. Finally, a case study of a chemical product manufacturer was discussed to show that the near-optimum safety stock level could be determined and that the customer satisfaction level could be estimated using this proposed simulation framework. Although the solution obtained from this framework is not optimal in a mathematical sense, the implementation is much more practical compared to directly solving the multi-stage stochastic program.

A large number of other studies of inventory control and management under risk/uncertainty are available in the literature. One of the most recent reviews in this research area is that of Gümüs and Güneri (2007), to which readers interested in supply chain risk management using inventory management are referred, as this aspect is not central to the present study.

Although increasing an inventory level in a supply chain can mitigate some risks, such a strategy incurs additional storage costs. Another approach that can be used effectively in mitigating supply chain risks is that of developing a more robust supply chain network using an appropriate network design approach. Specifically, in our research, the robust network design approach will be emphasized as a key risk mitigation strategy. The following discussions highlight studies related to supply chain network design in the presence of risks that have been done in the past.

Supply chain network design includes the tasks of selecting suppliers, production facilities, and markets; establishing capacity, and making order allocations. In the literature, a large number of research efforts have been devoted to methods for a whole network design and to methods for only two echelons in a supply chain. Examples of the works that aim to solve a problem in a two-echelon supply chain include Tomlin (2006) and Chopra et al.(2007). Tomlin (2006) studied a problem with one firm, two suppliers, one product type, and the percentage of uptime of known suppliers. The two suppliers consisted of one reliable but more expensive supplier and another one unreliable but cheaper supplier. This study showed that when a firm is risk-neutral, a supplier's percentage uptime and the nature of the disruption risks (frequent but short versus rare but long) significantly affect the optimal strategy for selecting the suppliers and determining order allocation. In particular, as a disruption becomes less frequent but longer, the firm should increase the quantity ordered from the reliable supplier (sourcing mitigation strategy). On the other hand, if the disruption is more frequent but short, the firm should buy more material and carrying inventory from the unreliable but cheaper supplier (inventory mitigation strategy). Last of all, it was shown that contingent rerouting of the supplier nodes is advantageous when disruptions are rare (contingency strategy). Chopra et al.'s (2007) study work presents the same result; specifically, that firms should order more from a reliable (expensive) supplier when the risk to the supply chain risk is rare but longer, but more from an unreliable (cheap) supplier when the supply chain risk is frequent but shorter. However, in contrast to Tomlin's models, Chopra et al.'s model assumed that the suppliers' uptime or uncertainty are unknown when placing an order with the first supplier and also it was required that the firm must reserve a maximum order size with the reliable supplier at a given reservation price.

Another approach is to consider a supply chain network as a whole when designing network configurations. This network design approach is usually achieved by the efficient use of mathematical programming models. Snyder et al. (2005) reviewed mathematical programming models that can be used in planning for disruptions in supply chain networks. The models were divided into two main parts: design models and fortification models. The design models are those used to design networks from scratch, whereas the fortification models are used to modify existing supply chain networks in order to render them more able to withstand impending risks. Furthermore, for each category of models, there are facility location models and network design models. The distinction between these subcategories is that there are transshipment nodes in the network design models, but no transshipment nodes in the facility location models. In addition, for each subcategory of the models, problems were further divided into expected cost-based optimization and worst case-based cost optimization models. These contributed a total of eight optimization models reviewed in that work. A more specific work in supply chain network design under risk is that of Gaonkar and Viswanadham (2007), which considered a supply chain with multiple facility plants and suppliers. This supply chain was subject to deviations (operational risks) and disruption risks. The objective was to minimize the risks' overall impact on costs. This study proposed an integer quadratic programming model for partner selection to minimize the overall cost impact from deviation in supplier costs (operational risk), and a mixed integer programming model for partner selection to minimize the overall impact of disruption risks. The advantage of the models in (Gaonkar and Viswanadham, 2007) is their simplicity and, as a result, they could be solved using Microsoft Excel with the Solver add-in. Santoso et al. (2005) proposed a more complicated model for supply chain network under uncertainty. The main objective of their work was to develop a more efficient algorithm for solving a supply chain network design problem with uncertain demands, supplies, capacity, and processing/transportation costs. The authors first formulated the problem as a two-stage stochastic program, and then tried to solve it using modified versions of the well-known Sample Average Approximation (SAA) and Benders decomposition methods. The computational results showed that a large-scale network design problem can be solved efficiently using their proposed algorithms. Numerous research studies in the area of network design/facility location models under uncertainty and risk are available in the literature. A more comprehensive review of this research area can be found in Snyder (2006), which includes reviews of stochastic location problems and robust location problems.

In our work, we mainly focus on the risk mitigation approach based on a supply chain network design under uncertainty as it is part of the reconfigurable supply chain, a proposed supply chain information system architecture in this work. In the next section, we focus on a review of models for supply chain network design under risk.

3.4 Models for supply chain network design under risk

A considerable body of research exists in the area of designing supply chain networks under risk or uncertainty. This work can be divided into four main subareas according to their solution approaches: analytical models, mathematical programming models, simulation models, and Artificial Intelligence (AI)–based techniques. This section presents a brief review of the literature on analytical, simulation, and AI approaches and offers a more intensive review of mathematical programming models, as the latter is a focus of the present study.

In the analytical solution approach, the problems under consideration usually involve only two echelons of a supply chain. These two echelons could be a manufacturer and suppliers, a manufacturer and retailers, or a distributor and retailers. For example, Tomlin (2006) focused on supplier selection and order allocation by considering a situation in which a firm could source from two suppliers, one perfectly reliable (but expensive) and another one unreliable (but cheaper). Subject to disruption risks, the unreliable supplier would be unable to supply any material if a disruption were to occur. To integrate the risk into the models, the disruption was modeled as a discrete-time Markov chain process. The author also introduced three types of risk management strategies: inventory mitigation (the firm sourced from the unreliable supplier and also carried some inventory), sourcing mitigation (the firm sourced from the reliable supplier), and contingency rerouting (the firm sourced from the unreliable supplier, but when this supplier was down the firm rerouted its order to the reliable supplier). Given a set of assumptions, i.e., (1) the firm is risk-neutral, (2) demand is deterministic, and (3) the unreliable supplier has infinite capacity, the study developed an optimal ordering policy for the firm, which stated the optimal base-stock level of inventory and how many units to order from the reliable supplier when the unreliable supplier is unable to fulfill the firm's inventory needs. In addition, the study also determined the optimal sourcing strategy; that is, how many units to order from each reliable and unreliable supplier when the sourcing mitigation strategy is appropriate. Based on its analytical models, the study presented the optimal disruption management strategy determined by the duration of the disruption and the reliability of the unreliable supplier. This disruption management strategy can follow one of the four strategies, i.e. inventory mitigation, sourcing mitigation, or contingency rerouting, or the combination of them. Another supplier selection and order allocation study that is closely related to (Tomlin, 2006) was discussed in (Chopra et al., 2007). Chopra et al. (2007) studied the impact of differentiating recurrent (operational) risk from disruption risk. The problem was set with one firm and two suppliers (also, one reliable and another unreliable). The authors showed that bundling the two uncertainties with recurrent and disruption risks could lead to the underutilization of a reliable source and the overutilization of less reliable sources. In addition, as the probability of disruption grows, we should mitigate the risks by ordering more material from the reliable supplier and less from the cheaper suppliers. To demonstrate the importance of a distribution center in reducing the demand risk faced by risk-adverse retailers, Agrawal and Seshadri (2000) considered a single-period inventory problem with one manufacturer and multiple risk-adverse retailers. With the addition of an intermediary, i.e., a risk-neutral distributor, between the manufacturer and the retailers, the retailers will order more products to maximize their utility functions; hence, they will increase supply chain efficiency, i.e., the expected profit of the whole chain. The demand risk, therefore, will be mitigated. This risk mitigation could be achieved if the distributor were to create contracts that shared the demand risk among the distributor and the retailers. The study showed that having been offered the optimal contract menu, the retailers chose the contracts that maximized their own utilities and the distributor's expected profit. As we can see, analytical models have usually been used in relation to small-scale problems, i.e. those with only 2 echelons of a supply chain. For large-scale problems, therefore, the other alternative must be considered.

Supply chain network design using mathematical programming models including Integer Programming (IP), Mixed Integer Programming (MIP), and Stochastic Programming (SP) has been widely studied. And, problem formulations have been conducted for various types of problems, and a number of efficient algorithms have been developed to solve such large-scale problems more accurately and quickly. Since network design using the mathematical programming method is the core subject of this research, an intensive review of this topic follows.

One of the early studies in the area of supply chain design is that of Eppen at al. (1989). The authors discussed the development of a capacity-planning model at the General Motors Company (GM), and considered a multiple-product, multiple-plant, and multiple-period capacity planning problem. The fundamental issue was that of determining the appropriate type and level of production capacity at each of several locations while there were uncertainties regarding the demand for and sale prices of the company's products. Firstly, the scenario planning analysis was applied to GM's strategic capacity planning procedure. In this case, there were three possible scenarios for demand uncertainty and also three possible scenarios for price uncertainty. Based on these generated scenarios, the capacity-planning problem was formulated for $H_s + 1$ possible configurations ($h = 0, 1, \ldots, H_s$) of the capacity at site k ($k = 1, 2, \ldots, K$) as follows:

$$Max \sum r_{ikhmt} p_{mt} x_{ikhmt} + \sum r_s \tau_{is} p_{mt} z_{imt} - \sum F_{kht} w_{kht} - \sum C_{kht} y_{kht} \qquad (3\text{-}4)$$

s.t. Demand constraints

$$\sum_{k=1}^{K}\sum_{h=1}^{H_k} x_{ikhmt} + z_{imt} = D_{imt} + \sum_{j=1}^{N}\tau_{ji}z_{jmt} \quad ;$$

$$i = 1, \ldots, N \quad m = 1, \ldots, M \quad t = 1, \ldots, T \tag{3-5}$$

Capacity constraints

$$\sum_{i=1}^{N} a_{ikh}x_{ikhmt} \leq G_{kht}y_{kht} - L_{kht}w_{kht} \, ;$$

$$k = 1, \ldots, K \quad h = 1, \ldots, H_k \quad m = 1, \ldots, M \quad t = 1, \ldots, T \tag{3-6}$$

Each site must always be in some configuration:

$$\sum_{h=0}^{H_k} y_{kht} = 1 \quad ; \qquad k = 1, \ldots, K \quad t = 1, \ldots, T \tag{3-7}$$

Retooling forcing, later periods:

$$y_{kht} - y_{kh,t-1} \leq w_{kht} \quad ; \qquad k = 1, \ldots, K \quad h = 0, \ldots, H_k \quad t = 1, \ldots, T \tag{3-8}$$

Retooling forcing, first period:

$$y_{kht} \leq w_{kht} \quad ; \qquad k = 1, \ldots, K \quad h = 0, 2, \ldots, H_k \tag{3-9}$$

Retool at most once:

$$\sum_{n=0}^{H_k}\sum_{t=1}^{T} w_{kht} \leq 1 \quad ; \qquad k = 1, \ldots, K \tag{3-10}$$

Shutdown forcing:

$$w_{k0t} \leq y_{k0t} \quad ; \qquad k = 1, \ldots, K \quad t = 1, \ldots, T \tag{3-11}$$

Define FXCOST:

$$FXCOST = \sum_{k=1}^{\lambda}\sum_{h=0}^{H_k}\sum_{t=1}^{5}(F_{kht}\,w_{kht} + C_{kht}y_{kht})$$

Define scenario contribution margin:

$$scp_m = \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{h=1}^{H_k} r_{ikhmt}\,x_{ikhmt} + \sum_{s=1}^{2}\sum_{i=1}^{N} r_s\,\tau_{is}z_{imt} \quad ;$$

$$m = 1, \ldots, M \quad t = 1, \ldots, 5 \tag{3-12}$$

Define downside risk of scenario sequence i, j, k, l, m:

$$d_{ijklm} \geq \tilde{z} - (scp_{1i} + scp_{2j} + scp_{3k} + scp_{4l} + scp_{5m} - FXCOST \quad ;$$

$$all\ i, j, k, l, m \tag{3-13}$$

Define expected downside risk:

$$ZRISK = \sum_{i=1}^{M}\sum_{j=1}^{M}\sum_{k=1}^{M}\sum_{l=1}^{M}\sum_{m=1}^{M} d_{ijklm}p_{1i}p_{2j}p_{3k}p_{4l}p_{5m} \tag{3-14}$$

Risk reduction constraint:

$$ZRISK \leq \Phi ZRISK_0 \tag{3-15}$$

Non-negativity and integrality:

$$x_{ikhmt}, z_{imt}, d_{ijklm} \geq 0; \quad y_{kht}, w_{kht} \in \{0, 1\} \tag{3-16}$$

where

- D_{imt} = demand for product i under scenario m in period t
- τ_{ij} = proportion of the unsatisfied demand for product i transferred to product j. Demand is unsatisfied if there is insufficient capacity to meet it. For i = j, $\tau_{ij} = 0$
- G_{kht} = production capacity of site k in configuration h, period t
- a_{ikh} = amount of capacity required to produce one unit of product i at site k under configuration h
- L_{kht}= capacity lost during period t, if site k is retooled into configuration h during period t
- F_{kht} = fixed cost of changing the configuration at site k from 1 to h (h = 0, 2, . . ., H_k) in period t (h = 1 is the initial configuration)
- T = total time periods, i.e., T = 5
- C_{kht} = fixed cost per period of having site k in configuration h during period t, e.g., the cost of having the plant and labor available
- r_{ikhmt} = variable contribution margin from producing (and thus, selling) one unit of product i at site k with configuration h under scenario m in period t
- r_s = variable contribution margin (this contribution accrues when demand for a product that can be produced in this model is unsatisfied and is transferred to product s, a product that cannot be produced in this model)

- p_{mt} = probability that scenario m will occur in period t
- x_{ikhmt} = amount of product i produced (and sold) at site k in configuration h under scenario m in period t
- y_{kht}= 1 if site k is in configuration h in period t, 0 otherwise
- z_{imt} = amount of unsatisfied demand for product i in scenario m and period t
- w_{kht} = 1 if site k is retooled into configuration h (h = 0, .. , H_k,) in period t, 0 otherwise
- $\tilde{z}$= point below which the downside risk function begins to take on a positive value
- $ZRISK_0$ = expected downside risk for the unconstrained optimal solution
- Φ = parameter between 0 and 1 used to tighten the risk constraint
- FXCOST = present value of all fixed costs associated with a solution, i.e., the sum of fixed costs from the configurations selected and the retooling costs
- scp_{tm} = present value of the contribution margin from scenario m and period t
- d_{ijklm}= deviation of the profit below z, if scenarios i, j, k, l and m occurred in that sequence
- ZRISK = expected value of the downside risk

This model not only maximizes the profit of a firm, but also minimizes the expected downside risk. The problem was solved using the linear mixed-integer programming in the LINDO software. The solution obtained from this model had significant influence on GM's capacity planning strategy.

Snyder et al. (2005) reviewed several mathematical models that we can use in planning for a supply chain subjected to disruptions at the supplier level. Of these, the model of most interest to our research is the reliability fixed-charge location model, a facility location model that minimizes an expected cost. This model is based on an uncapacitated fixed-charge location problem (each facility has unlimited capacity and an annual fixed cost f_j). There were a fixed set I of customer locations and a set of J of potential facility locations. In addition, each of these facilities had a fixed probability of failure equal to q, and when the facility failed, it could not supply any products at all to its customers. Once a facility had failed, a customer's order would be routed to the next other available facility. The next closest facility to the initial facility was at the distance level 1, the 2nd closest facility was at distance level 2, and the ith closest facility was at level i, where there were totally r levels of distance. The reliability fixed-charge location problem can be stated as below.

$$Min \sum_{j\epsilon J} f_i X_j + \sum_{i\epsilon I} \sum_{r=0}^{|J|-1} \left[\sum_{j\epsilon J\{u\}} h_i d_{ij} q^r (1-q) Y_{ijr} - h_i d_{iu} q^r Y_{ius} \right] \quad (3\text{-}17)$$

$$s.t. \quad \sum_{j\epsilon J} Y_{ijr} + \sum_{s=0}^{r-1} Y_{iur} = 1 \ ; \quad \forall i \in I, r = 0, \ldots, |J|-1 \quad (3\text{-}18)$$

$$Y_{ijr} \leq X_j \ ; \quad \forall i \in I, j \in J, r = 0, \ldots, |J|-1 \quad (3\text{-}19)$$

$$\sum_{r=0}^{|J|-1} Y_{ijr} \leq 1 \ ; \quad \forall i \in I, j \in J \quad (3\text{-}20)$$

$$X_j \in \{0,1\} \ ; \quad \forall j \in J \quad (3\text{-}21)$$

$$Y_{ijr} \in \{0,1\} \ ; \quad \forall i \in I, j \in J, r = 0, \ldots, |J|-1 \quad (3\text{-}22)$$

where

- f_i = fixed cost of facility i if it is opened
- X_j = 1 if facility j is opened, and 0 otherwise
- h_i = quantity ordered by customer i
- d_{ij} = distance between facility j and customer i
- q = probability that a facility will fail
- q^r = probability that a facility which is the rth far from a customer will fails
- Y_{ijr} = 1 if customer i is assigned to facility j at level r
- d_{iu} = distance from outsource supplier to a customer
- Y_{ius} = 1 if customer i is assigned to outsource supplier u at level s

The objective function (3-17) is to minimize the sum of the fixed cost and the expected transportation and lost-sales costs. Constraints (3-18) require each customer i to be assigned to some facility at each level r, unless i has been assigned to the outsource supplier at level s < r. Constraints (3-19) prevent an assignment to a closed facility, constraints (3-20) prohibit a customer from being assigned to the same facility at more than one distance level. Constraints (3-21) and (3-22) are binary constraints. As we can see from the formulation above, this model is able to capture the expected transportation cost without using explicit scenarios to describe the uncertain event (disruption).

An alternative approach to modeling uncertainty in a supply chain network is the two-stage stochastic programming method. The stochastic programming method is widely acknowledged as a modeling tool for optimization under uncertain conditions. Basically, the algorithm of two-stage stochastic program problems can be divided into two parts. The first stage involves the strategic decision related to facility locations (usually modeled as 0-1 variables), and the second stage corresponds to decision making at tactical level, namely, production and distribution plans (usually modeled as continuous variables). Stochastic programming models are usually large-scale mathematical programs; therefore, the problems are usually decomposed into smaller problems and solved on that basis. Two widely used decomposition techniques are the Lagrangian relaxation method and Benders decomposition method.

Lucas et al. (2001) proposed a method for solving a capacity planning problem under uncertainty. Formulated as a two-stage stochastic supply chain, the problem was efficiently solved using the Lagrangian relaxation technique. The problem considered was a supply chain network with four major stages: production, packing, distribution centers, and final customers. The only uncertain parameter in this problem was customer demand (in this case, there are a hundred of demand scenarios). After scenario planning analysis had been conducted, the problem was formulated as a mixed-integer programming model in order to minimize the value of cost–

revenue. However, the authors argued that this type of problem is too large to solve using traditional algorithms. Therefore, the problem was reformulated as a two-stage stochastic program as follows:

$$Min\ Z = cx + \sum_s p_s f y_s \tag{3-23}$$

s.t.

$$Ax = b \qquad\qquad logical\ \ constraints \tag{3-24}$$

$$Bx + Dy_s \leq h \qquad \forall s \in \{1, \dots, S\} \qquad capacity\ constraints \tag{3-25}$$

$$Ey_s = d_s \qquad \forall s \in \{1, \dots, S\} \qquad demand\ constraints \tag{3-26}$$

$$x \in \{0,1\}^{n_1} \qquad y_s \geq 0 \tag{3-27}$$

where

- c = cost vector of building the facilities
- x = vector of facility decision variables (x = 1 if the facility is built, 0 otherwise)
- p_s = probability of scenario s
- f = vector of costs of production, packing, ordering, transportation, and shortage
- y_s = vector of quantities of second stage variables, i.e., production, packing, ordering, transportation, and shortage quantities for scenario s
- b = vector of limit number of facilities
- h = vector of capacities of facilities
- d_s = vector of customer demand for scenario s
- n_1 = total number of facilities
- A, B, D, and E = matrices of the corresponding data

However, to render the computation more tractable, the scenario analysis and Lagrangian relaxation technique were applied to the above problem. Therefore, the problem was reformulated and decomposed as follows:

$$Min\ Z = (c + \lambda B)x \tag{3-28}$$

s.t. $Ax = b$ (3-29)

$$x \in \{0,1\}, \lambda \geq 0 \quad (3\text{-}30)$$

$$Min\ Z = (f + \lambda D)y \quad (3\text{-}31)$$

s.t. $Ey = d_s$ (3-32)

$$y \geq 0, \lambda \geq 0 \quad (3\text{-}33)$$

The first part of the problem above is a pure integer problem for a fixed value of λ and the second part is a larger linear programming problem, where λ = the Lagrangian multiplier and the other notations are similar to Equations (3-23)-(3-27). Based on the model stated their work (equation (33)-(36) in (Lucas et al., 2001)), the author developed approximation methods to solve the problem and compared the results of the proposed method (Lagrangian relaxation) with those of the direct integer programming method.

Another decomposition method that can be used to solve a two-stage stochastic program is Benders decomposition method. Santoso et al. (2005) proposed a method for solving a two-stage stochastic program in designing a supply chain network under uncertainty. The supply chain considered consisted of multiple suppliers, facilities, warehouses, and customers. The two-stage stochastic problem can be formulated as follows (Santoso et al., 2005):

$$\text{Min}_y \quad f(y) = c^T y + E[Q(y,\xi)] \quad (3\text{-}34)$$

$$s.t. \quad y \in Y \subseteq \{0,1\}^{|P|} \quad (3\text{-}35)$$

Where $Q(y,\xi)$ = the optimal value of the following problem:

$$\min \quad q^T x + h^T z \quad (3\text{-}36)$$

$$s.t. \quad \sum_{i \in N} x_{ij}^k - \sum_{l \in N} x_{jl}^k = 0 \ ; \qquad \forall j \in P, \forall k \in K \quad (3\text{-}37)$$

$$\sum_{i \in N} x_{ij}^k \geq d_j^k \ ; \qquad \forall j \in C, \forall k \in K \quad (3\text{-}38)$$

$$\sum_{j \in N} x_{ij}^k \leq s_i^k \ ; \qquad \forall i \in S, \forall k \in K \quad (3\text{-}39)$$

$$\sum_{k \in K} r_j^k \left(\sum_{i \in N} x_{ij}^k\right) \leq m_j y_j \ ; \qquad \forall j \in P \quad (3\text{-}40)$$

$$x \in R_{+}^{A \times K} \tag{3-41}$$

where

- A Latin character with subscript represents a single variable and a Latin character without subscript represents a vector variable
- c = vector of investment cost for building facilities
- y = vector of binary variables, $y_i = 1$ if a facility is built, or 0 otherwise
- Y = set of all possible processing nodes in a supply chain
- $Q(y, \xi)$ = optimal operating cost of a supply chain given the network configuration y, and uncertain parameter ξ
- ξ = vector of random parameters, i.e., ξ is a vector of (q, d, s, m)
- q = matrix of unit processing and/or transporting cost of a product under uncertainty, q_{ij} is the unit processing cost of a product processed at node i and sent to node j
- x = matrix of quantity of products, x_{ij}^{k} is the quantity of product k from node i to node j
- h = matrix of unit shortage cost, h_{ij} is the shortage cost of a product processed at node i and sent to node j
- z = matrix of shortage quantity, z_{ij} is the shortage quantity of product from node i to node j
- N = set of all nodes, S = set of supply nodes, P = set of processing nodes, and C = set of customer nodes. Therefore, N = S U P U C.
- A = set of arcs
- K = set of product types
- d_j^k=uncertain demand of product k at node j
- s_i^k= uncertain supply of product k at node i
- r_j^k= per-unit processing requirement of product k at node j

- m_j = uncertain capacity of node j

Then the evaluation of (3-34) was done by an approximation method called "Sample Average Approximation" method and the optimization was completed using the Benders decomposition algorithm. In addition, to obtain more computational efficiency, several techniques were integrated to accelerate the Benders decomposition algorithm; for example, Knapsack inequalities were added to the problem constraints, a heuristic algorithm was used to find the upper bound, and the cut in the algorithm was strengthened. The computational results showed that these acceleration techniques can certainly improve the accuracy and computational time of the existing algorithm.

Simulation- and AI-based techniques also form the basis for other operations research models for supply chain network design. Chan and Chan (2005) demonstrated how to evaluate supply chain network designs using simulation models. These supply chain networks were operated under demand uncertainty. The authors first considered three different categories of supply chain network configurations. Each configuration consisted of four echelons: supplier, manufacturers, retailers, and customers. The first category was called the "Inter-organizational supply chain model" in which there was only one supplier, manufacturer, retailer, and customer for each product type. The second category was the "Network supply chain model". This network supply chain allowed the collaboration among suppliers and also among customers. Therefore, for example, if the raw material is short at one supplier, such supplier can request additional material from another available supplier and then transfer this material to the manufacturer. The third category was the "Regional clustering supply chain model" in which for each product type, there could be more than one supplier, manufacturer, retailer, and customer. Hence, the material availability would be increased in this supply chain model. Simulation models were run for each category of supply chain network configuration, and then their performance in regard to average order lead time, transportation cost, resource utilization, and inventory level was evaluated. The results suggested that no network configuration dominates all other configurations. The appropriate strategy is to investigate the behavior of the supply chain considered and then select the most appropriate configuration based on the results of the related simulation. Deleris and Erhun (2005) published a related study, which is similar to that of Chan and Chan (2005) in the sense that both studies used simulation models to evaluate supply chain configuration in the presence of risks; however, in Deleris and Erhun (2005) the risk was coming from the supply side of the supply chain. The authors developed a supply chain model with four types of possible risks at the supplier level: employee strikes, shortage of components, political instability, and disruption caused by hurricanes. It was further assumed that once the risk event had taken place, the corresponding node would be down and unable to supply the material. Production volume losses and also financial performance were then evaluated for the predetermined supply chain configurations. Although this work can be implemented as a tool to evaluate a supply chain network design subject to disruption risks, it failed to address issues related to operational risks. Therefore, additional work is needed to investigate the effects of the operational risks to improve the robustness of a more realistic supply chain. Another study that not only used a simulation model to evaluate a supply chain configuration but also to directly design the configuration was proposed by Mele et al. (2007). This study used an agent-based simulation approach to design a supply chain network by considering both its configuration and inventory control strategy. The risk in this supply chain model was caused by demand uncertainty generated by sampling from the Monte-Carlo simulation. The first step of this framework was to select candidates for supply chain configuration. Then, total profit yielded by each configuration was evaluated for different inventory control strategies. The proposed framework can be divided into two parts: Agent-Based Supply Chain (ASC) simulation and tactical-level optimization. The tactical-level optimization used the Genetic Algorithm (GA) to optimize the total profit of each supply chain configuration, and then the operational parameters obtained from such optimization were sent to the agent-based simulator. This simulator then computed the expected profit from each supply chain configuration. This expected profit kept improving until the optimality criterion was reached, at which point the simulation terminated. The resulting inventory control policy constituted the best policy for each particular configuration. Then the configuration that gave the highest expected profit was selected as optimal. However, one drawback of the simulation approach is its sub-optimality property. Therefore, to improve the quality of the solutions, the optimality/convergence criterion must be emphasized when designing a supply chain network. Pai et al. (2003) discussed how a Bayesian network together with Fuzzy Logic can be used to assess business risk and evaluate the robustness of a supply chain network. Teuteberg (2007) used a neural network to determine the probability of risk occurrence and also identify the risk level at each node in a supply chain network. However, difficulty still exists, since the neural network approach requires a large amount of training data. Therefore, this approach may not be appropriate for risk events that rarely occur, such as those associated with disruption risks.

3.5 Industrial best practices in supply chain risk management

Due to the increasing vulnerability of supply chain networks, firms and organizations must of necessity focus their attention more on supply chain risk management. Since the fire at the plant of its only supplier in 2000 that rendered the company unable to offer any products to its customers (Norrman and Jansson, 2004), Ericsson has developed and implemented supply chain risk management strategy to achieve a more robust supply chain network. The risk management process at Ericsson includes risk identification, risk assessment, risk treatment, and risk monitoring. Figure 3-1 illustrates the company’s supply chain risk management process.

Figure 3-1: Risk management process at Ericsson (Norrman and Jansson, 2004)

In the risk identification process, firstly, the business process flow was developed, and then critical or risky suppliers/service providers were identified. In addition, Ericsson also tries to understand the impact of the risks by looking at how the duration of an accident (or other disruption event) will affect its supply chain performance. In general, the longer the incident is, the more severe the impact. Next, the risk assessment process determines the effects of each risk to each business unit. To facilitate this process, Ericsson developed its own tool—the Ericsson Risk Management Evaluation Tool (ERMET)—to evaluate and quantify the impacts of the risk based on their consequences and probabilities. Then, based on the result from the ERMET, Ericsson converts this operational loss into financial loss called "Business Interruption Value (BIV)." The value of BIV is directly related to the severity of risk. Next, in the risk treatment process, appropriate mitigation strategies are developed to reduce the effect of the supply chain risks. Last of all, Ericsson also continues monitoring the result of their risk mitigation strategies. If the problem still exists, further mitigation actions may be required. Contingency plans are an important part in the risk management process at Ericsson. Ericsson has divided the contingency plan into three steps: (1) response plan to assess the level of containment and to control activity, (2) recovery plan to resume malfunctioning operations or processes, (3) restoration plan to plan and implement the full-scale business operations again and to allow the organization to return to normal service level.

The other best practice in supply chain risk management is the total cost supplier selection model used at General Electric (GE) Wiring Devices Company, as discussed in Smytka and Clemens (1993). GE defines their supplier selection process in terms of three main elements: (1) risk factors,(2) business desirable factors, and (3) measurable cost factors. The risk factor stage is a go/no-go decision making process. At this stage, a cross-functional team of ten GE members work together to assess the risk exposure resulting from each available supplier according to the risk factors that have been previously identified. The aim of this stage is to determine whether or not such a supplier is qualified as GE's supplier in terms of risk management. Once the supplier receives a "go" evaluation at the first stage, the business desirable factors stage is used to evaluate supplier performance on some attributes that are difficult to express financially, such as quality, productivity, or miscellaneous capabilities. Each of the attributes is scored, and the total score for each supplier is calculated. Those suppliers that have an acceptable total score will be considered further in the next stage of the selection process, which focuses on measurable cost factors. Given a short list of supplier candidates, the measurable costs are calculated for each candidate. These measurable costs consist of external costs (e.g., price of supplies, discount term, and ordering cost) and internal costs (e.g., GE's inventory cost, cost of supply shortage, and nonconformance cost resulting from the suppliers). The supplier ultimately selected through this process will be the one that according to GE's criteria has the lowest measurable cost. However, the drawback of this strategy is that it tries to select only one supplier for procurement—a risky move in the face of supplier disruption risks. Therefore, in such a case, more than one supplier with a low measurable total cost could be selected to improve the robustness of the supply chain network.

Another industrial example related to procurement risk management is Hewlett-Packard's (HP) supplier portfolio approach, as discussed in (Billington, 2002). In this approach, HP developed a method for selecting suppliers to procure materials using the well-known financial portfolio model that considers both expected return and variance. Not only does HP allocate its order to several suppliers, but it also uses different procurement contracts for different suppliers. Different types of contracts result in different risk-taking parties in a supply chain. For example, in a long-term structured contract, the buyer commits to buying a specified number of units over a specified period of time; hence, the buyer is taking the risk of purchasing units even if they do not meet the buyer's needs at a future point. On the other hand, in an unstructured contract, the buyer provides a forecast and expects the supplier to supply an uncertain quantity without changing the price. However, usually the buyer will be charged for a fixed percentage of the component price. In this case, the supplier bears the risk since the buyer may or may not buy the units. Mainly, HP's portfolio procurement process consists of the following stages: (1) Strategy and Governance: HP specifies the acceptable level of risk and the expected return; (2) Source Solicitation: potential suppliers are invited to submit bids for sourcing; (3) Portfolio Evaluation: potential suppliers are evaluated against the original parameters set for the portfolio during the strategy and governance stage; (4) Contract Execution: a contract agreement is created for each selected supplier; and (5) Contract Monitoring: the suppliers are monitored for compliance and also for their value to the portfolio. If the portfolio fails to deliver the expected value, the procurement team must re-evaluate the portfolio and adjust it as needed.

The literature we have discussed so far are concerned with supply chain risk management. In the following section, we will explore the literature related to Service-Oriented Architecture (SOA) and its applications to supply chains as a means for deploying our reconfigurable network in practice.

3.6 Service-Oriented Architecture in supply chain modeling

Real supply chain networks are complex, dynamic, and heterogeneous. Therefore, it is crucial that enterprises or nodes in a supply chain must be seamlessly integrated to allow precise information flow among parties in the network. Service-Oriented Architecture (SOA) is an emerging software architecture that can be adopted to model, control, and automate business processes and operations in a supply chain network. In addition, it facilitates efficient supply chain coordination in such a complex, dynamic, and heterogeneous environment based on its principles of loose coupling, discoverability, interoperability, and autonomy.

Any application in a business network consists of several functions performed by different parties in the network. For example, consider material procurement in a supply chain, which starts with an inventory control person requesting a specific quantity of raw material from a procurement department. Then, a procurement person searches for and contacts suppliers for quotations of that order. Next, the suppliers respond to quotation by offering unit price, quantity, and delivery date to the procurement person. Based on these offers received from the suppliers, the procurement person purchases the orders that maximize his utility function. From the perspective of SOA, each function in this procurement process can be viewed as a service performed by node(s) in the network. That is, in the case of this procurement, the process consists of supplier searching, requesting for quotation, purchasing services performed by the procurement person and offer creations performed by the suppliers. Figure 3-2 illustrates this procurement process modeled using a service-oriented paradigm.

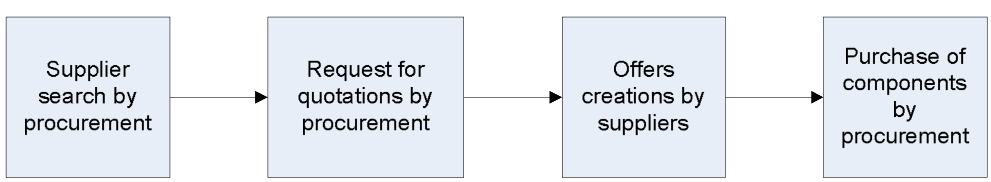

Figure 3-2: Procurement process as a service composition

SOA is basically a modeling approach that views a business application as a composition of services such as the procurement process discussed above. Erl (2005) defined SOA as "a form of technology architecture that adheres to the principles of service-orientation. When realized through the Web services technology platform, SOA establishes the potential to support and promote these principles throughout the business process and automation domains of an enterprise." The following are the main properties of SOA as defined by Erl (2005):

1. Loose coupling: services maintain a relationship that minimizes dependencies and only requires that they retain an awareness of each other.
2. Service contract: Services adhere to a communications agreement, as defined collectively by one or more service descriptions and related documents.
3. Autonomy: Services have control over the logic they encapsulate.
4. Abstraction: Beyond what is described in the service contract, services hide logic from the outside world.
5. Reusability: Logic is divided into services with the intention of promoting reuse.
6. Composability: Collections of services can be coordinated and assembled to form composite services.
7. Statelessness: Services minimize retaining information specific to an activity.
8. Discoverability: Services are designed to be outwardly descriptive so that they can be found and assessed via available discovery mechanisms.

Although not explicitly stated, interoperability can be achieved via the service-oriented design as well, as interoperability is a by-product of the loose coupling and abstraction. In our work, a supply chain information system with the eight properties stated above is developed to enable network reconfiguration ability when a risk event appears imminent or has occurred.

To implement SOA in any business or supply chain network, standardization is required to promote seamlessly integrated enterprises in the network. Several integration technologies have been available in the past, for example, Distributed Component Architecture (DCOM), Remote Method Invocation (RMI), Common Object Resource Broker Architecture (CORBA), and Web Services (WS). Among these technologies, Web services developed by the World Wide

Web Consortium (W3C) (http://www.w3.org/) is the most appropriate technology for enabling the development of the SOA due to its supporting characteristics for dynamic supply chain management and the considerable support of major commercial software vendors. Therefore, in our research, the realization of SOA using Web services technology will be studied in terms of its implementation in a service-oriented supply chain network.

Typically, the SOA system based on Web services technology consists of three main elements: service requester, service provider, and service directory. The service requester requests a service by sending a message to the service directory. The service directory contains a list of available service providers that can fulfill the request from the service requester. Once the service directory has received a message, it searches through the list of providers, and then returns a set of associated providers with their contact information to the requester. After the requester has received the information of the associated providers, it sends messages to such providers to initiate negotiation for the services desired. Figure 3-3 represents communication among elements in Web services technology (Dietrich et al., 2007).

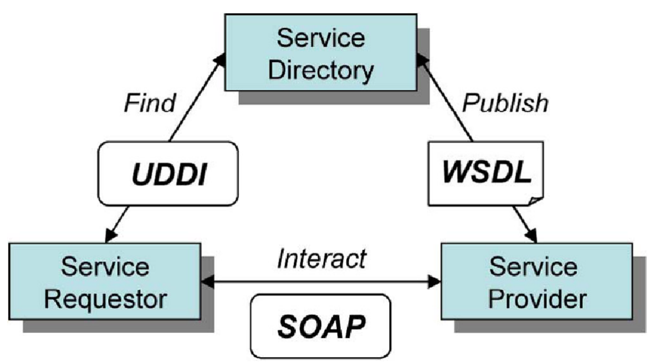

Figure 3-3: Communication among Web services elements (Dietrich et al., 2007)

To facilitate the communication between elements in the SOA system, some standardization is required:

- UDDI (Universal Description Discovery and Integration): UDDI specification defines a way to publish and discover information about Web services. This information about the services will be available in the service directory.
- WSDL (Web Services Description Language): WSDL provides a way to describe and publish services in the service directory.
- SOAP (Simple Object Access Protocol): SOAP provides a protocol for exchanging XML-based messaging format between the services.

These three standardizations are widely used among SOA researchers and practitioners currently. Both Microsoft .NET framework and Sun's J2EE also support the use of these three standards. At present, too, other Web service standards continue to be released to academia and industry, for example, WS-Reliable messaging, WS-Security, and WS-Eventing.

In the literature, a growing number of projects and research methodologies have been proposed for modeling supply chains using SOA. Zhang et al. (2006) proposed a modeling framework for developing a service-oriented architecture in a supply chain, providing a modeling framework at an abstract level of SOA. This study divided the modeling task into two models: the Meta-model of service component and the Meta-model of supply chain management systems. It used Unified Modeling Language (UML) to model and describe information related to the services in the Meta-model of the service component, such as identification, service contract, and service content. On the other hand, in the meta-model of supply chain management systems, three types of service components take on the roles of supplier, manufacturer, and retailer. Then the two Meta-models were integrated to achieve the SOA-based supply chain model. Kart et al. (2006) developed the MIDAS (Managing Integrated Demand and Supply) system that provides a Web services-based SOA for managing a supply chain network. This study considered a three-level supply chain network with several customers, one manufacturer, and several suppliers. The manufacturer in the MIDAS system consists of the following components: the Material Manager, the Orders Manager, the Database Manager, the Registry (Directory service), and the Quotes Manager. In addition, this SOA system was run as a simulation model to make decisions regarding the quantity of material to order and how to allocate the order among the selected suppliers. Qiu et al. (2007) presented their project focusing on the exploration of a practical approach to enable a software platform of a service-oriented supply chain, and proposed a service-oriented integration architecture model for general enterprises or organizations. This architecture consists of five layers: Enterprise business application, Business process (business function services), Generic services (standard connectivity), Process services (services and services compositions), Rules and logics (computing operations). The first two layers are business operations from the system perspective, and the last three layers are the value-adding processes from the execution perspective. These layers from the system perspective and execution perspective are integrated using the Service-oriented integration technology developed on the IBM Websphere platform. Using the integration model proposed in Qiu et al. (2007) together with XML-based messaging, a model for a service-oriented supply chain consisting of Manufacturing, Transportation, Warehousing, and Retailing systems can be developed successfully.

Dietrich et al. (2007) developed a service-oriented architecture for a mass customization system in the shoe industry. The main objective of this work was to develop a seamless and common platform that can be adopted by supply chain partners to promote interoperability. In this mass customization, customers participated at the stages of fabrication and assembly. The authors first created an actor model to describe the interaction among the elements of the mass customization system, including the customers, vendor, configurator, retailer, producer, suppliers, and logistic service provider (LSP). Then the services that are performed by the actors are derived, for example, the product configuration service can be derived when a customer accesses the configurator on the website and tries to customize the product. Next the product model is constructed. This product model describes components in the product, for example, which components are required to build this product or which options are available to build this product. The customization process can be completed through the use of this product model. Then, the SOA model is created. This model is presented as a UML diagram showing the relationship among the components and services in the system. The system is implemented using Java and the UDDI, WSDL, and SOAP standards. Shen et al. (2007) proposed agent-based service-oriented integration architecture to leverage manufacturing scheduling services on a network of virtual enterprises. A unique property of their approach is that the job-scheduling process is completed over the Internet through negotiations among agent-based Web services. This software platform is the combination of a multi-agent system and Web services technology. By encapsulating services in the agents, the capabilities of Web services technology can be enhanced including dynamic business formation and effective selection of services, service dynamic composition, flexible cooperation strategies, and semantic and ontology abilities. The model in their work can be divided into two levels, i.e. design level and implementation level. At the design level, the Web services were encapsulated as agent models so that each agent functions on behalf of a Web service. At the implementation level, UDDI, WSDL, and SOAP provide capabilities of discovery, service deployment, and communication, respectively. The final result is an agent-based service-oriented manufacturing system with inter-enterprises, resource sharing, and a dynamic environment. Lastly, Kumar et al. (2007) conducted a survey and empirical analysis to address the question of whether the SOA actually improves supply chain performance. The data was gathered from supply chain practitioners, and the Ordinary Least Squares (OLS) regression method was used to determine the relationship between the adoption of SOA and the firm's revenue. The results showed that SOA does indeed improve supply chain revenue by efficiently integrating the customer into the supply chain. However, this improvement was based on the performance on the revenue side only. Further analysis is required to understand the effects of SOA on other aspects of supply chain performance such as cost reduction and customer satisfaction.

In our research, the design of a robust supply chain network will be achieved by using a stochastic programming approach. However, this network configuration can resist negative impact from only one risk at a time and, hence, this alone may not be sufficient in practical supply chain risk management. By the use of SOA, the supply chain can be automated such that it can reconfigure itself as different kinds of risk events are determined to be more or less likely. Therefore, adopting SOA would promote the ability of a network to withstand several types of risk events should they impact a dynamic supply chain.

Chapter 4

Methodology for Network Reconfiguration: An Overview

The basic idea of network reconfiguration for a supply chain subject to risk is that of enabling a supply chain network to adjust its configuration when a risk event likely to impact the network is imminent. Such an ability to adjust itself would make the network configuration more capable of protecting itself against an expected risk and, hence, the total cost of the supply chain can be reduced compared to the total cost of a traditional supply chain. To develop a reconfigurable supply chain network, three components are required: (1) an uncertain parameter database, (2) a stochastic network optimization module, and (3) a network configuration controller. Figure 4-1 presents a schematic overview of this reconfigurable supply chain network.

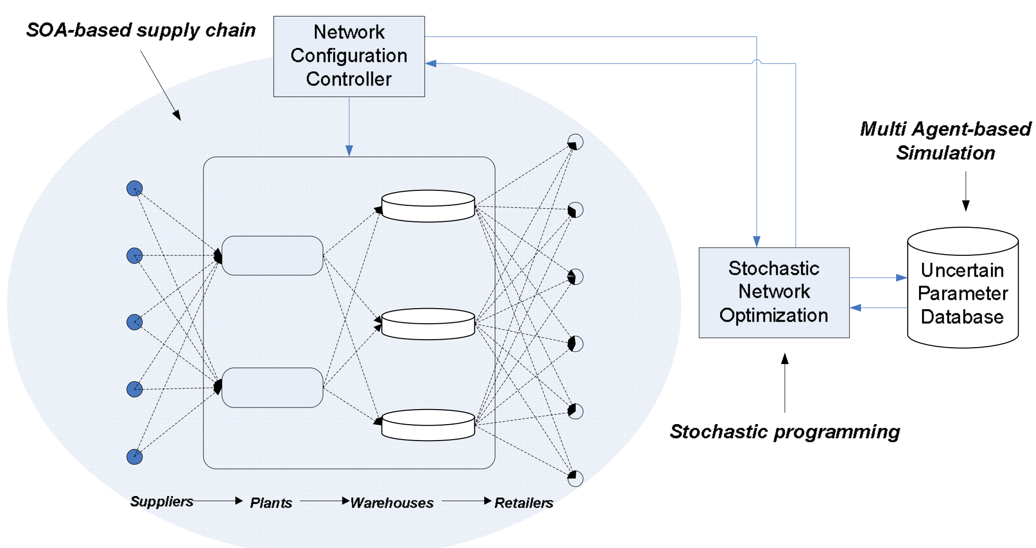

Figure 4-1: Overview of a reconfigurable supply chain

From Figure 4-1, in order to reconfigure a supply chain network, the stochastic network optimization module requests data about network design parameters from the uncertain parameter database. These data include probability distributions of shipping costs, customer demands, supply availabilities at supplier nodes, and facility capacities, all of which are obtained by running simulations on a Multi-Agent based simulation model. After receiving the data about these probability distributions, the stochastic network optimization module formulates a problem of supply chain network redesign in a two-stage stochastic programming using these probability distributions. It then solves the problem to obtain a solution to long-term facility location decisions, i.e., plant and warehouse locations. Then the list of the selected plant and warehouse locations is sent to the network configuration controller. Based on the given list of plants and warehouses, this network configuration controller in turn formulates a daily (short-term) supply chain network configuration in a Linear Programming (LP) model to determine supplier and customer portfolios, as well as order and product allocations on a daily basis. This daily supply chain network configuration is sent to the selected plants and warehouses to form an optimal supply chain network for each day of operation. Finally, to allow all the supply chain nodes that are operating on diverse enterprise software platforms to work together, the Service-Oriented Architecture (SOA) is applied to the information system architecture design of this supply chain. Figure 4-2 shows a detailed procedure of the network reconfiguration subject to risk.

According to Figure 4-2, the procedure starts by detecting or predicting risk events that could occur in the supply chain network in step 1. It is assumed that a supply chain planner is able to detect the possibility of a risk event based on personal experience and/or some forecasting techniques. In step 2, once the possibility of risk has been detected, the stochastic network optimization module will be triggered to start a network redesign task. This network optimization module redesigns the supply chain network using a two-stage stochastic programming model by receiving the probability distributions of the network design parameters from the uncertain parameter database and a list of all available nodes in the environment from a service registry (not shown in Figure 4-1).

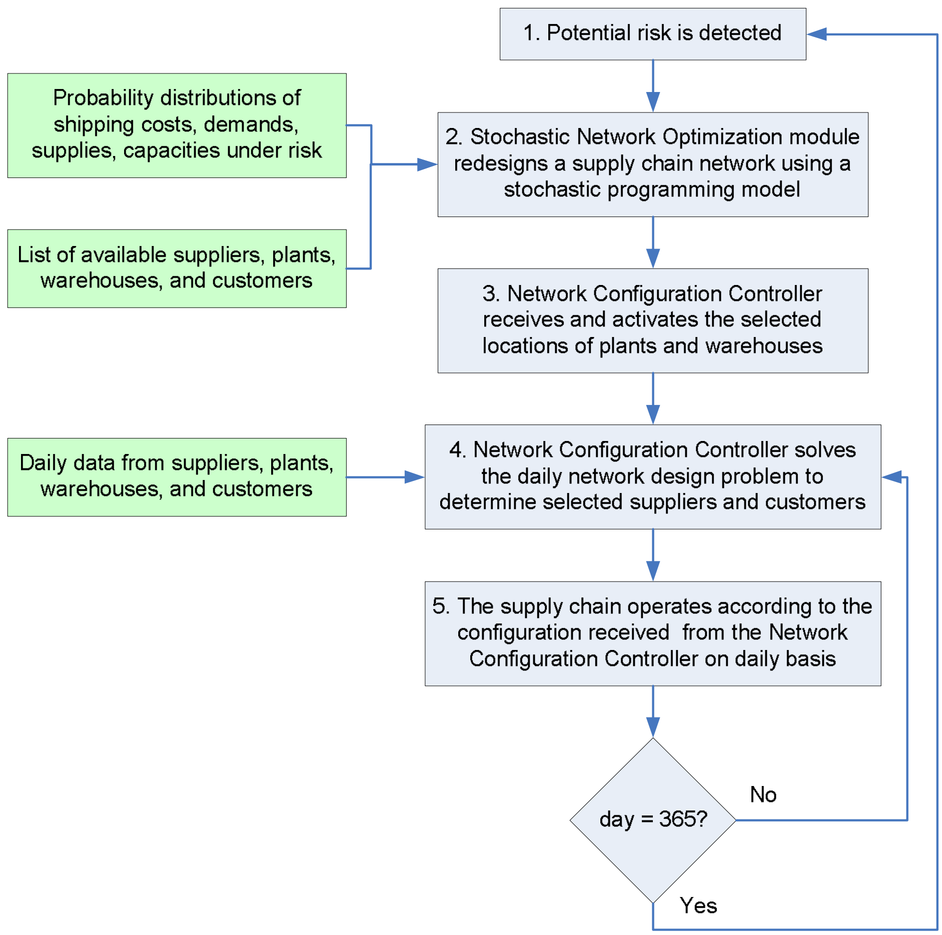

Figure 4-2: Network reconfiguration procedure

The solutions obtained in step 2 are locations of the selected plants and warehouses for the supply chain. In particular, this network configuration at the plant and warehouse layers is a long-term decision and will be not be changed frequently due to the high level of investment necessary for building or acquiring facilities. In step 3, the network configuration controller receives this facility location data from the optimization module and activates operations at the selected facilities. In step 4, the network configuration controller receives real time or daily data of the network design parameter from the previously selected plants and warehouses and from all available suppliers and customers. Next, it determines the supplier nodes from which to procure raw materials and the customer nodes to which to sell products using a daily network optimization model which is an easy Linear Programming (LP) problem. Basically, the network configuration controller determines the configuration at the supplier and customer layers for a short time period, perhaps, on a daily basis. This frequently changed configuration is possible because the cost of switching supply chain partners at the supplier and customer layers is low. Next, in step 5, the network configuration controller sends messages to the previously selected plants and warehouses instructing them to contact the newly selected suppliers and customers as trading partners of the supply chain. Lastly, the supply chain planner checks whether a criterion for reconfiguring the supply chain has been met; e.g., in Figure 4-2, the planner checks whether 365 days have passed since the last network reconfiguration. If yes, the process starts again from step1; otherwise, the process returns to step 4. The final result is a reconfigurable supply chain network that can operate at a satisfactory level of performance in the presence of risk events.

In the following chapters, details regarding the uncertain parameter database, stochastic network optimization, and network configuration controller will be discussed.

Chapter 5

Understanding Risk Effects: A Multi-Agent-Based Simulation Approach

According to Figure 1-1 in Chapter 1, the first step in developing a reconfigurable supply chain network is to understand the effects of risks on a supply chain network. In this chapter, an approach used to categorize supply chain risks is discussed and also the effects of risk events are studied by collecting and analyzing the data of network design parameters, including shipping costs between nodes, customer demands, supply availabilities, and facility capacities, when a supply chain is subject to risk. These data are in turn fitted into the probability distributions to represent uncertainty in the parameters. Because a sufficient amount of associated data is required in order to develop the proper distributions for such network design parameters, the available historic data we have in hands may not be sufficient to represent due to the nature of supply chain risks that rarely occur in the real world. Alternatively, we can consider the use of a simulation model for generating and collecting the data that are needed for a distribution fitting task. The resulting fitted distributions are then stored in the uncertain parameter database as shown in Figure 4-1and integrated into the stochastic network optimization module in order to generate samples of the random network design parameters to the stochastic programming model accordingly.

In this research, a Multi-Agent-based Simulation (MAS) approach is used to simulate operations in a supply chain network. A typical supply chain network consists of several independent firms working together so that each can achieve its own objectives, the MAS approach is considered the most effective tool for modeling operations in a supply chain. In the MAS, each firm or node in the supply chain is modeled as an autonomous software agent that has its own beliefs and logic. In addition, the agents in the MAS model are also able to communicate with each other in order to complete required processes or tasks in the supply chain. These communications among the agents are standardized to ensure accurate communications among the agents. According to the advantages mentioned above, the MAS approach is used to observe the behavior of the supply chain network under risk conditions. The following sections discuss the risk categorization and the design and implementation of the MAS model. The methodology for the probability distribution fitting is then presented at the end of this chapter.

5.1 Risk categorization

In the literature, some researchers have categorized supply chain risks according to their respective frequency and/or duration (Chopra et al., 2007; Tomlin, 2006). The use of frequency and duration are sufficient in these studies because they considered only 2-layer supply chains, e.g., suppliers and a manufacturer, or a distribution center and customers. However, the present research focuses on a whole supply chain network that contains 4 layers of nodes. Therefore, location of the risk is also important in modeling it. The following are attributes used to categorize supply chain risks in this research:

- Risk frequency: there are three levels of frequency: high (6 times/year), medium (4 times/year), and low (2 times/year).
- Risk duration: there are three levels of duration: long (13–15 days), medium (6–8 days), and short (0–2 days).
- Location: a risk can occur in one of the following layers in the supply chain: supplier, plant, warehouse, or customer layer.

Since we have 3 levels of frequency, 3 levels of duration, and 4 locations of risk, we will have a total of 36 types of risks. Therefore, when performing experiments to study the effects of risk events, 36 experiments will be carried out. It is necessary to note here, too, that the attributes listed above are based on a general assessment of factors that need to be accounted for. These attribute values, therefore, should be adjusted appropriately when applied to a real-world supply chain.

5.2 Design and implementation of the Multi-Agent-Based Simulation model

The design of the MAS model in this research is a combination of the Gaia methodology for agent-oriented analysis and its design concept (Wooldridge et al., 2000) and the Unified Modeling Language (UML) standard (http://www.uml.org/). The architecture of the MAS model derives from the Gaia methodology, such that each agent in the supply chain network performs one or more roles in the network. In addition, each role consists of one or more operations in order to complete the role assigned to the agent. Based on the architecture defined for the agents, roles, and operations, the UML diagrams are used to represent the low-level details of this MAS model. To complete the MAS design, we must first define the work flow chart of each agent. Then, we will convert the flow charts into business process flows using the sequence diagrams from the UML standard. Finally, based on the sequence diagrams, the agent roles and class representations are generated using the class diagram from the UML. The details of this MAS design process are discussed next.

The supply chain network under consideration is a network with four tiers of nodes. These consist of supplier, manufacturing plant, warehouse, and customer tiers. The MAS model was developed such that users can determine the number of nodes in each supply chain tier. In addition, the users can also define the configuration of each node in the supply chain via a configuration file. This configuration includes, for example, demand at each customer node, processing and transportation cost, facility capacity, safety stock level, and so on. Once the users have defined both the number of nodes in each supply chain tier and the node configurations themselves, the simulation can begin and each node in the network will operate independently and autonomously based on the concept of multi-agent system design.

Although this MAS model is intended to simulate the operations in a supply chain network with four tiers of nodes, another agent is required to develop the simulation model. This is the controller agent, which is responsible for monitoring and controlling the overall progress in the simulation model. Therefore, five classes of agents are created in this simulation model: controller agent, supplier agent, plant agent, warehouse agent, and customer agent.

As stated earlier, the first task in the MAS design is that of developing the work flow charts for the agents in the network. Thus, we first define the flow charts for the agents in the network as shown in Figure 5-1 to 5-5 below.

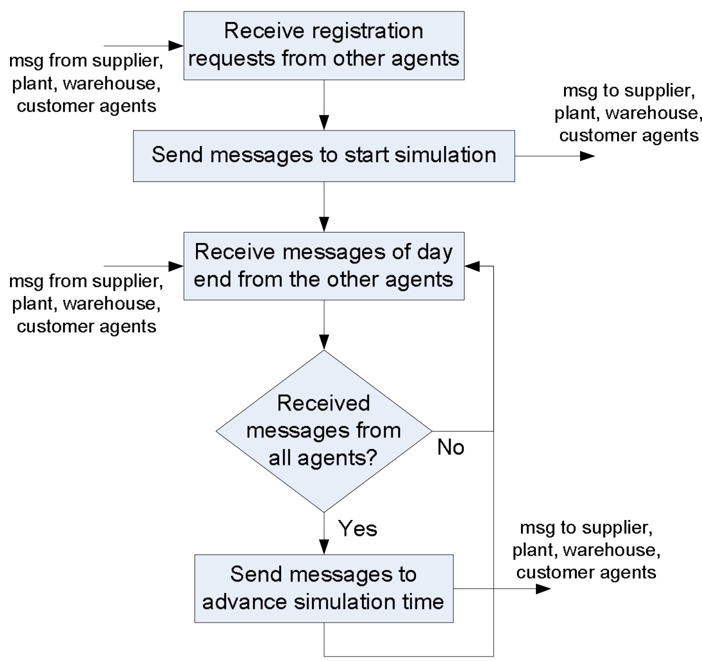

Figure 5-1: Flow chart for a controller agent

In Figure 5-1, the controller agent first receives registration requests from the other agents in the system. Then, it sends messages to all the registered agents to start their operations in the simulation. Each agent in the system operates its operations until all of the required operations are completed, then all the agents send "Day End" messages back to the controller agent to inform that they finished all tasks for today and are ready to advance to the next simulation day. Once the controller agent has received the messages from all the agents, it will advance the simulation time by one day.

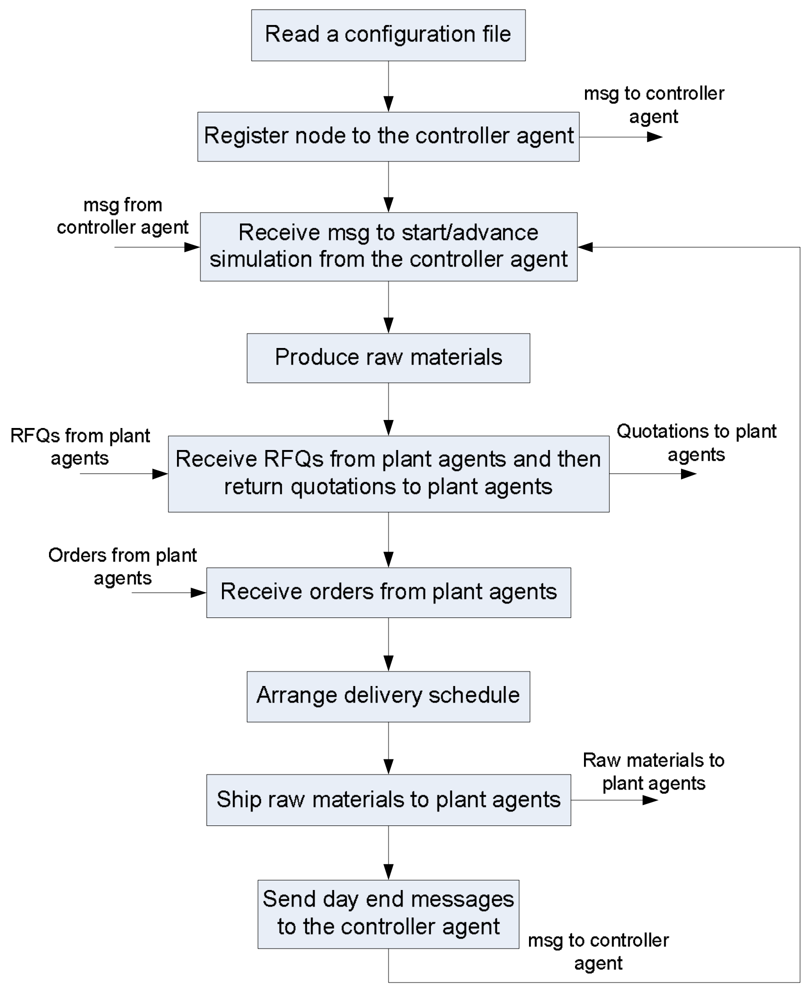

Figure 5-2: Flow chart for a supplier agent

Figure 5-2 illustrates the operations of the supplier agent. These operations include producing raw materials, receiving Requests for Quotations (RFQs) from plant agents, providing quotations to the plant agents, receiving orders from the plant agents, and delivering raw materials. Once all the operations are complete, the supplier agent sends the Day End message to the controller agent.

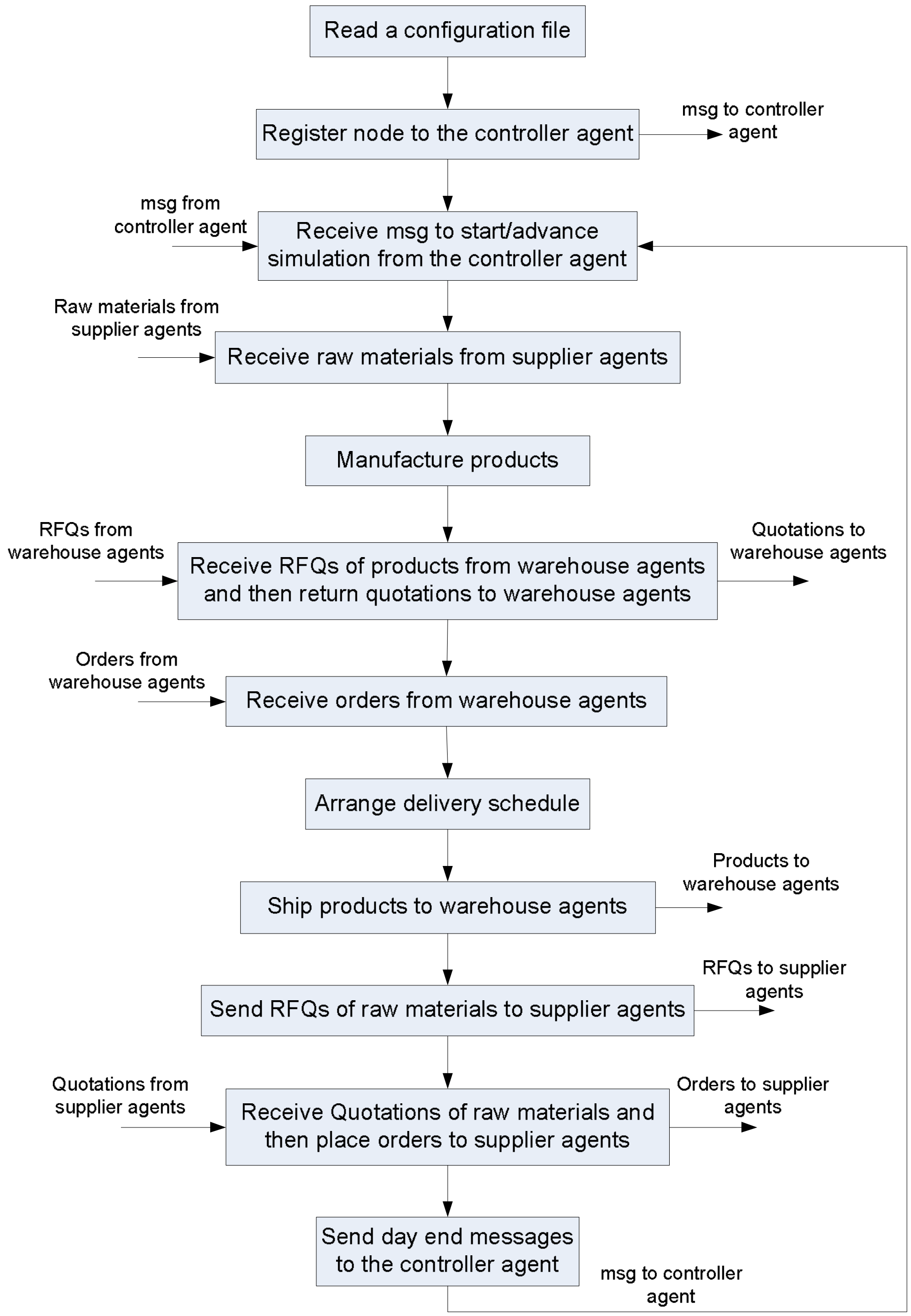

Figure 5-3: Flow chart for a plant agent

Figure 5-3 shows the operations of the plant agent. The operations of the plant agent consist of receiving raw materials from the supplier agents, manufacturing products, receiving RFQs from the warehouse agents, providing quotations to the warehouses, receiving orders from the warehouse agents, delivering products, sending RFQs for raw materials to the supplier agents, receiving quotations from the supplier agents, and placing orders to the supplier agents. After all the tasks are completed, the plant agent will send the Day End message to the controller agent.

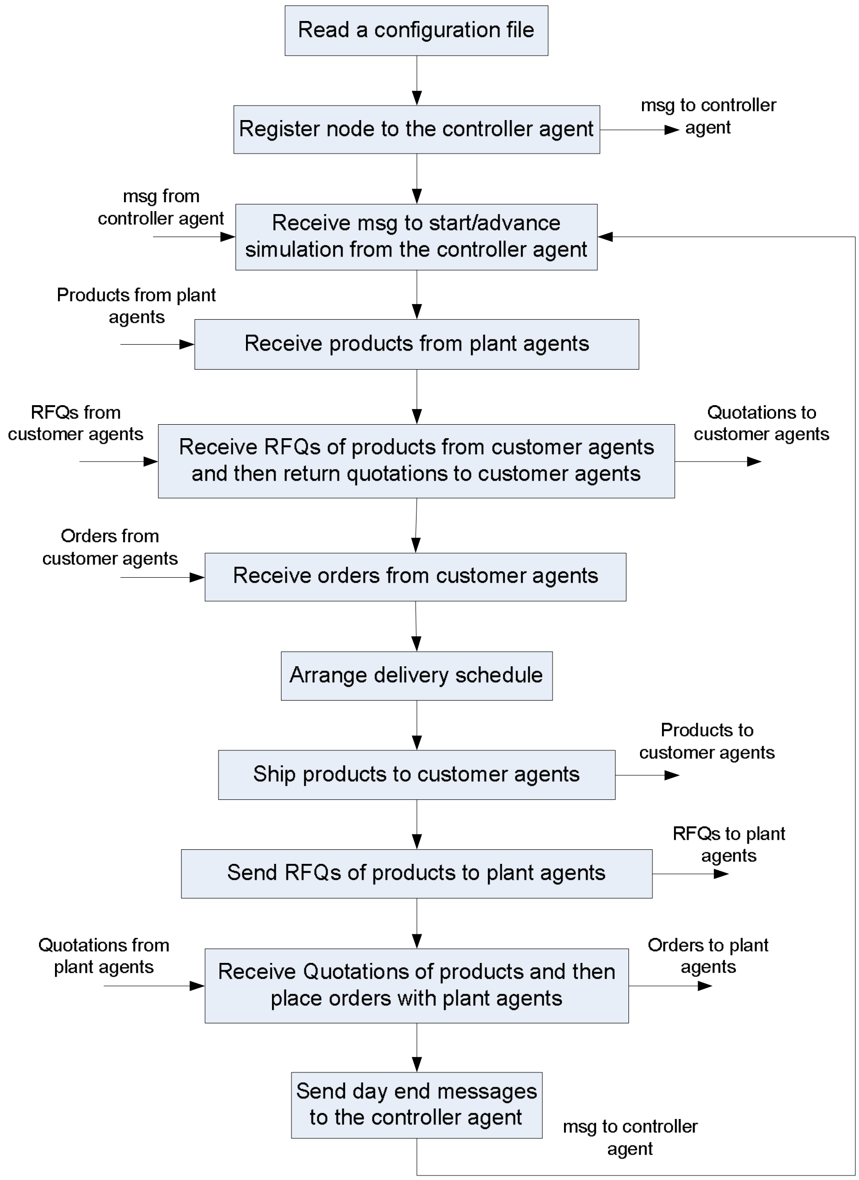

Figure 5-4: Flow chart for a warehouse agent

In Figure 5-4, the operations of the warehouse agent are much the same as those of the plant agent, except that the former has no manufacturing tasks. Similarly, after all the operations are completed, the warehouse agent will send the Day End message back to the controller agent.

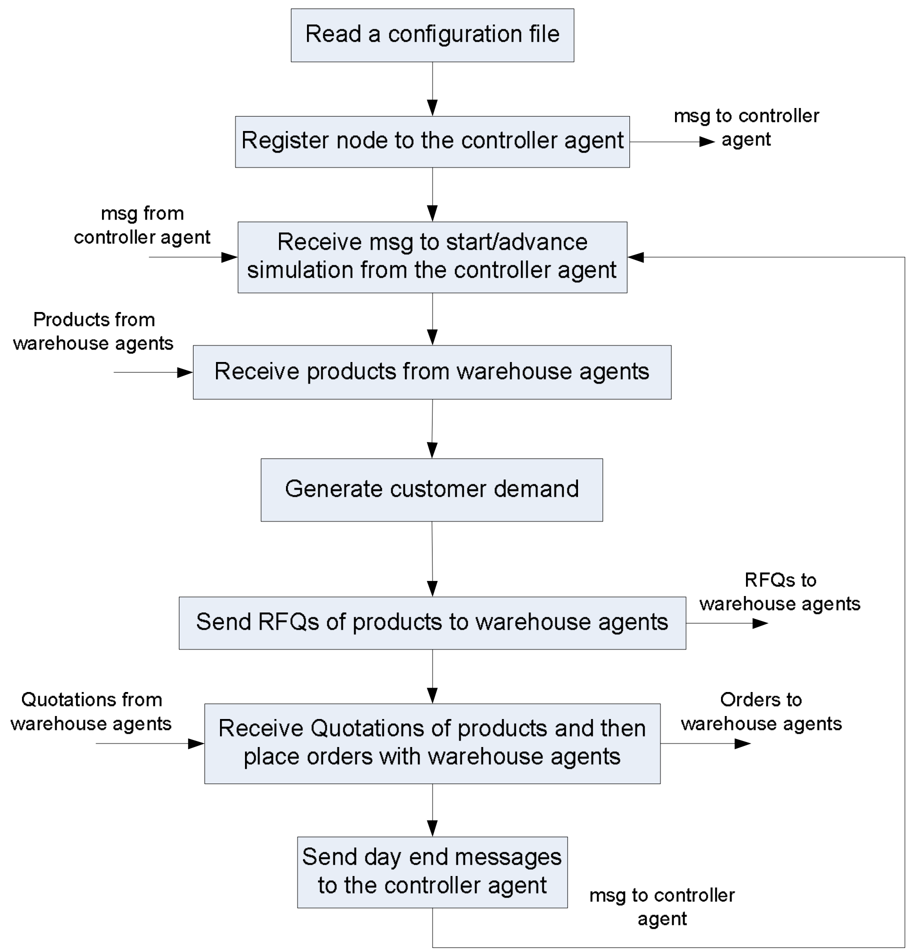

Figure 5-5: Flow chart for a customer agent

The customer agent's operations are shown in Figure 5-5. During the day, the customer agent is responsible for receiving products from the warehouse agents, generating random customer demand (normal distribution), sending RFQs of products to the warehouse agents according to the demand generated, receiving quotations and then placing orders to the warehouse agents. When all of these operations are finished, the customer agent sends the Day End message back to the controller agent.

Although most parts of this simulation model are flexible, such that users can determine the network configuration as they desire, some assumptions must still be made to simplify the implementation of the programming code. These assumptions are stated in Appendix A.

5.2.1 Presentation of the MAS model using UML

The flow charts in Figure 5-1through Figure 5-5 illustrate the operations of each agent in the simulation model. However, in order to develop the MAS model, design tools for representing the model are required. This research uses Unified Modeling Language (UML) as a tool in presenting the MAS model. In particular, the study references two kinds of diagrams from the UML standard: The first one is the sequence diagram and the second one is the class diagram. The sequence diagrams are used to represent business processes in the supply chain network, for example, raw material procurement, production planning, and product delivery. The class diagrams are used to model the agents themselves in the model. An example of a sequence diagram representing the raw material procurement process is shown in Figure 5-6.

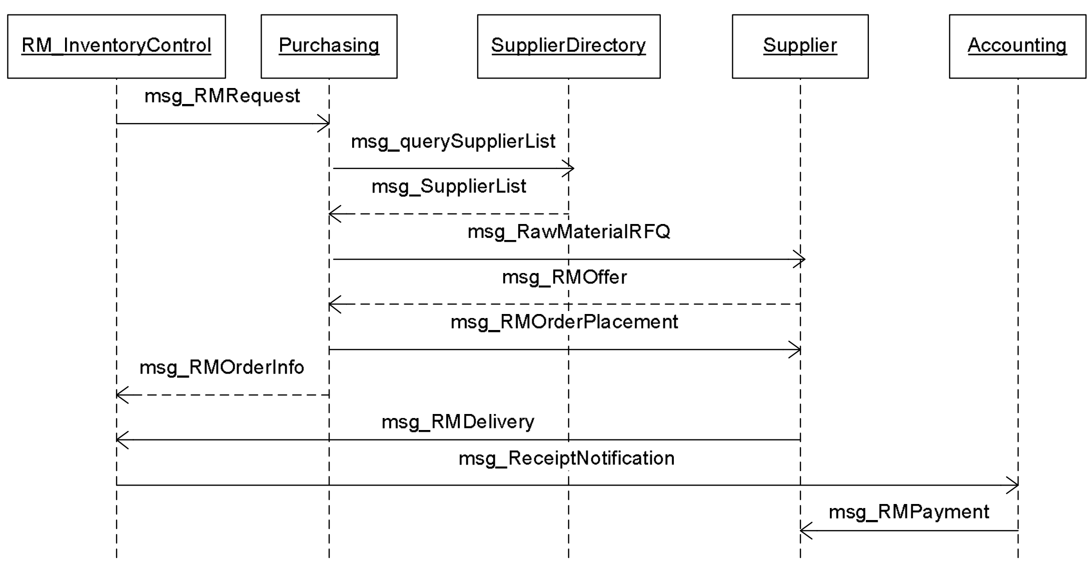

Figure 5-6: Sequence diagram for the raw material procurement process

In the diagram for the raw material procurement process, the vertical dash lines represent the roles of the agents and the role names are shown at the tops of the lines. In Figure 5-6, there are five roles, i.e., RM_InventoryControl, Purchasing, SupplierDirectory, Supplier, and Accounting. The RM_InventoryControl is the role of the plant agent that is responsible for controlling inward and outward raw material at the raw material storage of the plant agent. Purchasing role is another plant agent's role that is responsible for contacting supplier agents when the raw material is needed. The SupplierDirectory is the role that provides a list of available suppliers in the system. The Supplier is the role of the supplier agent and this role will respond to the plant agent if the request for the raw material is created. The Accounting role pays for the shipment when the plant agent receives the raw material from the supplier agent. The process of the procurement process in Figure 5.6 can be explained as follows: The RM_InventoryControl sends a message to the Purchasing when replenishment of the raw material is required. Then the Purchasing asks the SupplierDirectory for the list of the suppliers from which they can acquire the raw material. After the SupplierDirectory replies back with the supplier list, the Purchasing contacts the suppliers according to the list for procuring the raw material. The Supplier sends back an offer about the price and delivery date of the raw material to the Purchasing. If the Purchasing accepts the offer, an order is placed and the Purchasing also notifies the RM_InventoryControl about the purchase. Then, when the due date is reached, the Supplier delivers the raw material to the RM_InventoryControl. Once the raw material is received, the RM_InventoryControl notifies the Accounting to pay the Supplier for this purchase.

In this agent-based simulation model, five business processes—procurement, raw material inventory management, production planning, distribution planning, and delivery planning—are represented by five sequence diagrams. The procurement process has already been shown in Figure 5-6, and the other four processes are detailed in Appendix B.

Furthermore, each role of the agent also consists of several specific operations. For example, the Supplier role consists of manufacturing raw material, receiving the RFQs from the plant agents, providing offers or quotations regarding the RFQs, sending the offers to the plant agents, then receiving orders from the plant agents, delivering raw material, updating its inventory level, and finally receiving payments from the plant agents. The role model can be expressed in a class diagram; for instance, the role model for the Supplier role is shown in Figure 5-7 below, and the class diagrams for the other roles are provided in Appendix C.

Supplier
+manufactureRM() : int +receiveRFQ() +generateOffer() : object +sendOffer() +receivePurchaseOrder() +deliverRM() +updateInvetoryLevel() : int +receivePayment()

Figure 5-7: Class diagram for the Supplier role

According to the Gaia conceptual design principle of multi-agent based systems in (Wooldridge et al., 2000), each agent has at least one role. In other words, one agent can have one or more roles. In the present study, an agent is represented by a class diagram similar to the role models. The roles of each agent are stated in attributes of the class diagram. The class diagram for the plant agent is shown in Figure 5-8.

PlantAgent
-rawMatMonitoring : RM_InventoryMonitoring -rawMatControl : RM_InventoryControl -rawMatPurchasing : Purchasing -demandForecast : SaleForecasting -productMonitoring : FG_InventoryMonitoring_Plant -production : ProductionPlanning -productControl : FG_InventoryControl_Plant -firmAccounting : Accounting

Figure 5-8: Class diagram of the plant agent

The plant agent can perform the following roles: raw material inventory monitoring, raw material inventory control, purchasing, sales forecasting, finished goods inventory monitoring, production, finished goods inventory control, and accounting. These roles are stated as the attributes of the class diagram in Figure 5-8. Appendix D presents class diagrams for the other four agents in the system, the control agent, supplier agent, warehouse agent, and customer agent.

Now that the conceptual agent model is complete, we can implement this conceptual model as described in the next section.

5.2.2 Implementation of the MAS model

The design of the MAS model presented in Section 5.2.1 is implemented in JAVA on the Java Agent DEvelopment (JADE) platform. The JADE platform is a JAVA Application Programming Interface (API) that facilitates agent system developers in developing multi-agent systems conveniently in JAVA language (Bellifemine et al., 2003). The main advantages of JADE for developing this agent-based simulation are:

- Communication protocols: these communication protocols allow the agents to communicate with each other by the compliance with the Foundation for Intelligent Physical Agents (FIPA) standard (http://www.fipa.org/).
- Directory Facilitator (DF) service: this service provides a yellow pages service to all the agents in the system. The agents can search for the registered services that they need to contact via the DF service
- Agent management system: this system ensures that each agent will have a unique name in the system. Therefore, messages can be sent to the specified agents correctly.

Due to the above advantages, the MAS model is efficiently developed using JADE and JAVA language in this research. Next, experiments are conducted in order to observe the impact of supply chain risks on the network design parameters of the supply chain network. The details of these experiments are discussed in the next section.

5.3 Simulation experiments and results

After the completion of the design and implementation steps, experiments were run on the MAS model in order to observe the behaviors of the network design parameters when the supply chain is subject to risk. The simulation experiments were conducted on the same problem as discussed in Section 2.1.

5.3.1 Procedure and design of the experiments

According to the problem in Figure 2-1 and Table 2-1 in Chapter 2, the experiment can be performed according to the following steps:

1. Design the optimal supply chain network configuration with respect to the sum of fixed facility cost and operating cost using the Mixed Integer Programming (MIP) model shown in Equations (2-1)-(2-7).
2. Once we have obtained the optimal network configuration (shown in Figure 2-2), the MAS model is created based on this configuration. In addition, let all the model parameters be deterministic except customer demand at each node. The customer demands are random following normal distributions.
3. Run the simulation on this model to observe the data of costs, demands, supplies, and capacities at the nodes in this supply chain.
4. Insert only one risk at a time into the simulation model. Then, run the simulation again to observe the impact of the risk on the design parameter data. Repeat this step for all 36 types of risks as discussed in Section 5.1. Also record all the design parameter data as the inputs for the distribution fitting task later.

Therefore, according to the procedure stated above, we can perform the experiment as follows. For step 1, regarding the given network design problem, we solved the MIP problem from Equation (2-1)-(2-7) and the optimal configuration of the supply chain can be shown as in Figure 5-9.

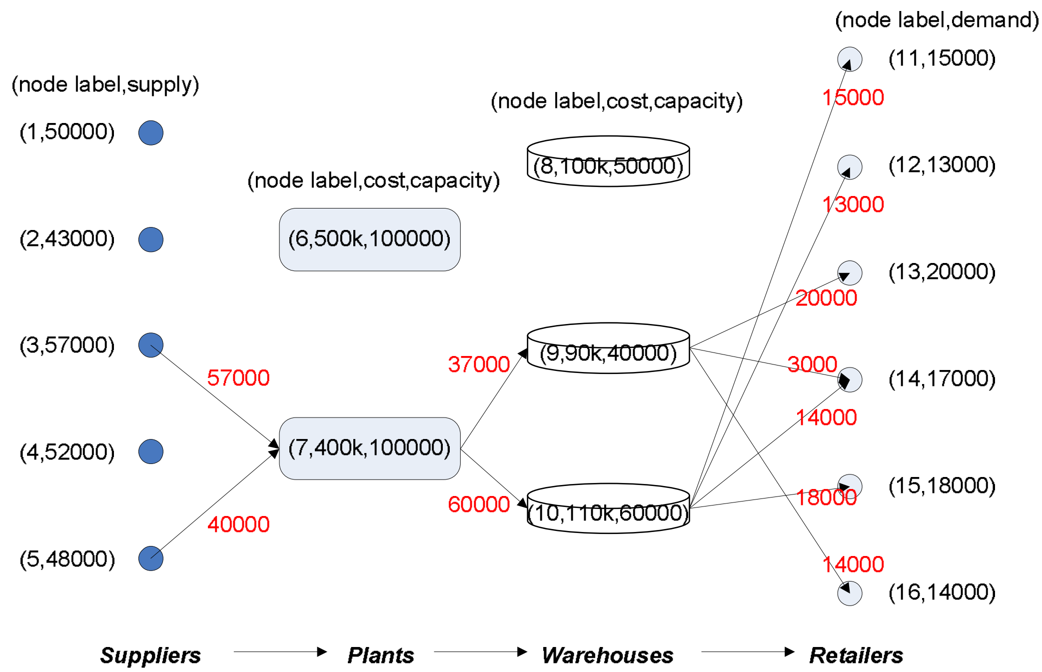

Figure 5-9: The optimal supply chain network configuration

Then, in step 2, we build the agent-based simulation model representing the configuration of the optimal supply chain network shown in Figure 5-9. However, the demand at each customer node will be normally distributed with a standard deviation equal to 20% of the demand in the deterministic problem. Next, in step 3, the simulation model is run for 1 year or 365 days of simulation time for 10 replications. Then, the data of the network design parameters are collected for the supply chain network without risk. In step 4, one type of risk is inserted into the MAS simulation to collect the data associated with the design parameters. Then this step is repeated for all 36 types of risks according to the discussion in Section 5.1. The data associated with the network design parameters are collected and fitted to the appropriate distributions as will be discussed in the next section.

5.3.2 Probability distribution fitting of the network design parameters

After the simulation experiments had generated a sufficient number of data points associated with the network design parameters, we can fit the probability distributions. At this stage, two alternatives can be considered in determining the probability distributions. The first one is to use the theoretical probability distributions such as normal distribution, exponential distribution, lognormal distribution, etc. The second approach is to develop an empirical distribution to each data type if none of the theoretical distributions can be appropriately fitted to the obtained data. The advantages of the theoretical distribution include:

1. Analytical models and solutions can be developed for some types of distributions such as exponential distribution.
2. It can generate all possible values of a random variable, as opposed to the empirical distribution, which generates bounded random values.
3. The empirical distribution may not be able to generate adequately good random stream that can represent a true underlying population due to sampling errors, which is not the case for the theoretical distributions.

Although theoretical distributions have several advantages over the empirical one, in this research, it is more convenient to model the network design parameters as the empirical distributions for the following reasons:

1. The data collected are not continuous and they include many jumps. Therefore, when fitting to any theoretical distribution model, the data cannot be statistically fitted to any standard theoretical distribution.
2. Some data associated with costs including supply, manufacturing, and warehousing costs appeared to be bounded. This is due to the fact that the MAS model assumed a lower

bound for the product price at each node; specifically, the lower bound is 70% of the mean price.

3. The advantage pertaining to analytical model development and solutions from the theoretical distributions may not be significant because the problem in this research will, however, be solved numerically on a computer.

An example of failure when trying to fit the experimental data into the theoretical distribution resulted from the MINITAB software is shown in Figure 5-10 and Figure 5-11.

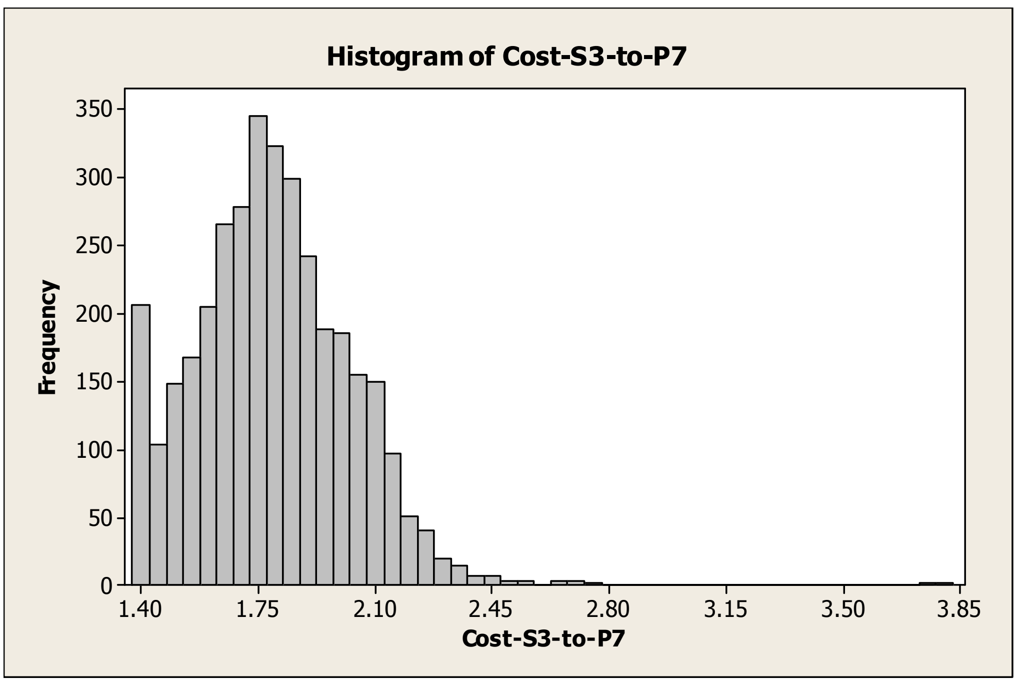

Figure 5-10: Histogram of the supplier 3 to plant 7 cost data

As Figure 5-10 shows, the data behavior is almost considered normally distributed. However, in the last column of the histogram, a large number of the cost data at 1.40 appears. This is because the lower bound of the product price at supplier 3 is 70% of 2.0, i.e., 1.40. Hence, the distribution of this cost data appears as seen in the figure. The test for normal distribution fitting also provides a similar result. The cost data set is rejected in the normal distribution test. Figure 5-11 shows the results of the normal probability plot.

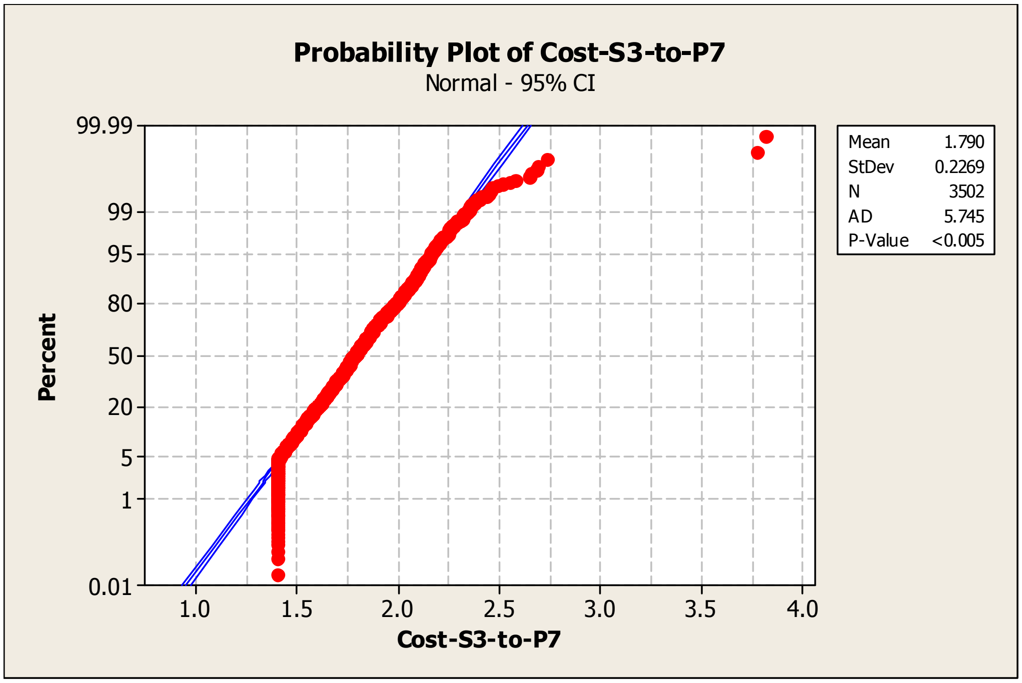

Figure 5-11: Normal probability plot of the supplier 3 to plant 7 cost data

According to Figure 5-11, we can visually reject the compatibility of the cost data considered and the normal distribution. In addition, the Anderson-Darling (AD) statistic of 5.745 and P-value of less than 0.005 at 95% confidence level further ensure that this data set does not follow the normal distribution by any means. Additionally, this behavior can be observed on all the design parameter data sets obtained from the simulation experiments.

Therefore, according to the discussion above, the empirical distribution is selected as an appropriate approach to representing the data of the network design parameter where the supply chain is subject to risk. The method for developing empirical distribution for each data set and integrating these empirical distributions into the network optimization module as introduced in Chapter 4 is explained next.

5.3.3 Developing the empirical distribution to the optimization module

The process for generating random samples to the stochastic network optimization module based on the empirical distribution is described in Figure 5-12.

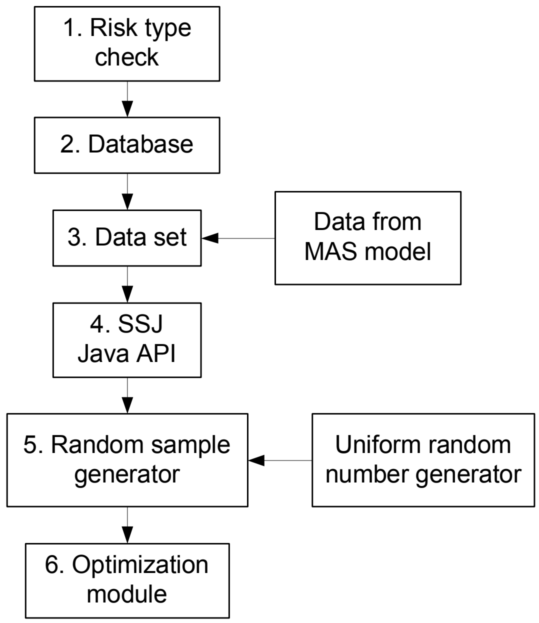

Figure 5-12: Random sample generator

As Figure 5-12 shows, the first step is to check the type of risk in the system using risk frequency, duration, and location. Once we know the risk type, in step 2, we can look it up in the uncertain parameter database. In step 3, the uncertain parameter database provides the location where the data set associated with each type of risk is located in our computer system (saved in .txt format). Then, once we know the location of the data set, the associated data set is read by the Java application in order to create the empirical distribution based on the given data set in step 4. Embedded within this application is an Application Programming Interface (API), which facilitates the creation of the empirical distribution. The API, referred to as the Stochastic Simulation in Java (SSJ) was developed at the Université de Montréal and is freely available for academic use (http://www.iro.umontreal.ca/). Then, in step 5, the uniform random sample is generated and fed into the random sample generator to output a sample that follows the desired empirical distribution to the stochastic network optimization module in step 6 accordingly.

At this point, we have successfully developed a system for generating the empirical probability distributions of the network design parameters. These empirical distributions serve two purposes: first, based on these distributions, we can visually learn about the effects of each risk on the supply chain network design parameters. Second, the empirical distributions can be used to generate random samples of the network design parameters as the inputs to the stochastic programming model in the stochastic network optimization module. However, these empirical distributions and the random sample generation are based on the data obtained earlier from the MAS model. Thus, when the configuration of the MAS is changed, the simulation experiment may need to be re-run to obtain more updated results and the new empirical distribution must be recreated. In other words, if the configuration of the real supply chain network is modified, it is recommended that the simulation experiment should be conducted again to ensure the validity of the empirical distribution as much as possible. In the next chapter, the two-stage stochastic programming model for a supply chain network under risk conditions and its approach to obtaining solutions will be presented.

Chapter 6

Supply Chain Network Redesign under Risk

The stochastic network optimization module is the second main component in the reconfigurable supply chain presented in Figure 4-1. It is responsible for redesigning the network configuration once any particular risk event is expected to occur in the environment. According to step 2 in Figure 4-2, two inputs are required to execute the stochastic network optimization module. The first one is a list of all available nodes, including available suppliers, potential customers, and current and candidate facilities in the supply chain. The second one that is necessary in the network optimization under uncertainty is data associated with probability distributions of the network design parameters, i.e., shipping costs, demands, supplies, and capabilities. For the former, the data can be obtained from a service registry that contains a list of all the nodes available. The details of this service registry will be explained in Chapter 7. The latter data is received from the uncertain parameter database with the Multi-Agent Simulation (MAS), as discussed in Chapter 5. In this Chapter, a two-stage stochastic model for a supply chain network redesign under risk that is solved in the stochastic network optimization module is discussed. In addition, the chapter discusses a solution approach and also a method for improving both solution quality and computational efficiency.

The basic idea of the stochastic network optimization module is that, once potential risk type k (ω_k) is detected in the system, the network optimization module sends requests to the uncertain parameter database and the service registry for the data required, as noted earlier. The data received from these two components are then used to formulate the stochastic programming model embedded in the stochastic network optimization module. As discussed in the previous chapter, the presence of risk type k (ω_k) in the system will cause randomness in some network design parameters, namely, production/shipping costs, customer demands, available supply quantities, and facility capacities. Therefore, to redesign the supply chain network, we need to capture these uncertain phenomena explicitly. In this case, a deterministic version of the supply chain network design model based on a traditional Mixed-Integer Programming (MIP) model, as presented in Equations (2-1)-(2-7) may not be appropriate for making decisions in this uncertain environment. On the other hand, a stochastic version should be considered. This research formulates a two-stage stochastic programming model to represent the problem of a supply chain network design under risk. In particular, since the present study takes the creation of network reconfiguration capability as its focus, the model formulated will be specific to a situation in which (1) the supply chain network configuration or physical infrastructure may already exists, and (2) an impending risk event necessitates that the supply chain network be reformulated to make it better able to withstand risk events. We will refer to this mathematical model as the Supply chain network Redesign under Risk (SRR) problem.

Our stochastic network optimization module is built based on three components:

- The two-stage stochastic programming model for the SRR problem: this component is responsible for redesigning the supply chain configuration by making optimal long-term decisions for new facilities locations including warehouses and plants. In this chapter, the mathematical programming model and solution approach for the SRR problem are discussed. In addition, three sampling techniques are applied to solve the problem to determine the best techniques (in terms of variance reduction and computational time) for problems of different sizes. Hence, the solution quality can be improved.
- Data input ports: the input ports receive data about the probability distributions of the random parameters and the available supply chain nodes from the uncertain parameter database and service registry, respectively.

- Data output ports: the output ports send requests for the required data to the parameter database and service registry when a risk is detected. Also, once the new configuration is determined, the data related to this new configuration is sent to the network configuration controller via the output port, as shown in Figure 4-1.

Details for the two-stage stochastic programming model for the SRR problem and its solution approach are given in the following sections.

6.1 A two-stage stochastic programming model for redesigning a supply chain network subject to risk

To formulate the specific stochastic programming model for the SRR problem, the following assumptions are required:

1. The supply chain network has four echelons (supplier, plant, warehouse, and customer echelons), and the product flows sequentially from the suppliers to the customers.
2. Only single type of product is delivered by this supply chain.
3. A supply chain planner can choose either to build or refrain from building its own plant at each candidate plant location. However, for warehouses, the planner has the options of building, renting, or refraining from using a warehouse at all at any particular location. We offer only two options for decisions regarding the plant and more for the warehouse because the technologies and processes required to manufacture products at a plant are more expensive and specific than those necessary for operations at a warehouse. Therefore, it is highly unlikely that a proper plant could be acquired at a reasonable rental cost. We have, therefore, limited the plant acquisition options, such that the firm can build its own plant or refrain from doing so.

4. At any specific location, a supply chain planner will choose to build a firm-owned warehouse only if the cost of doing so is lower than the annual cost of renting a warehouse multiplied by the firm-owned warehouse's expected useful life (years). For example, if the cost to the firm of building its own warehouse is $1 million and its useful life is 10 years, the planner will rent the warehouse only if the rental cost is less than $100,000, such that the warehousing cost for 10 years would be less than $1 million.
5. Any existing plant or warehouse that is not to be used should be closed, if the base operating cost is higher than the closing cost.
6. The management of the company has imposed limitation on reconfiguration budget. The cost of network reconfiguration cannot exceed this predefined limit.
7. The objective here is to determine the supply chain network configuration (warehouse and plant locations) in order to obtain the minimum expected facility and operation cost.

The optimization model of the SRR problem that follows is an extension of that in Santoso et al. (2005).

Indices and sets:

i, j = index of nodes in the supply chain

S = set of candidate suppliers

P = set of candidate plants

W = set of candidate warehouses

C = set of customers

$F = P \cup W$

$N_d = S \cup F \cup C$

A = set of arcs between nodes

Model parameters:

$y_{i,0}$ = 1 if we have a firm-owned facility i in the previous time stage, otherwise 0.

c_{oi} = cost of opening a firm-owned facility at node i

c_{ci} = cost of closing a firm-owned facility at node i

c_{ri} = cost of renting a warehouse at node i

c_{bi} = base cost of operating a facility at node i

l_i = useful life (years) of a firm-owned warehouse at node i

b = available budget for the reconfiguration project

$\boldsymbol{\xi}$ = random vector containing random variables **q**, **d**, **s**, and **m**; that is, $\boldsymbol{\xi} = (\mathbf{q}, \mathbf{d}, \mathbf{s}, \mathbf{m})$

ω_k = risk type k

$\mathbf{q}(\omega_k)_{ij}$ = random unit processing and transporting cost of product flowing from node i to node j under risk ω_k

g_j = unit penalty cost of unfilled order at customer node j

$\mathbf{d}(\omega_k)_j$ = random demand at node j under risk ω_k

$\mathbf{s}(\omega_k)_i$ = random available supply at node i under risk ω_k

r_j = per-unit capacity required when a product is processed at plant or warehouse j

$\mathbf{m}(\omega_k)_j$ = random capacity of facility j under risk ω_k

Decision variables:

$y_{i,1}$ = 1 if we will build an owned facility i, otherwise 0. Y_1 = vector of $y_{i,1}$ where i ∈ F

y_i = 1 if we will rent a warehouse i, otherwise 0. Y = vector of y_i where i ∈ W

x_{ij} = units of product flowing from node i to node j. X = vector of x_{ij} where i, j ∈ N_d

u_j = units of unfilled product at customer j. U = vector of u_j where j ∈ C

Problem formulation:

$$\text{Min } z = \sum_{i\in F} c_{oi}\left(y_{i,1} - y_{i,1}y_{i,0}\right) + \sum_{i\in F} c_{ci}\left(y_{i,0} - y_{i,1}y_{i,0}\right) + \sum_{i\in W} l_i c_{ri}\, y_i + \sum_{i\in P} c_{bi} y_{i,1} + \sum_{i\in W} c_{bi}(y_{i,1} + y_i) + E[Q(Y_1, Y, \boldsymbol{\xi}(\omega_k))] \quad (6\text{-}1)$$

$$s.t. \quad y_{i,1} + y_i \leq 1 \qquad \forall i \in W \quad (6\text{-}2)$$

$$\sum_{i\in F} c_{oi}\left(y_{i,1} - y_{i,1}y_{i,0}\right) + \sum_{i\in F} c_{ci}\left(y_{i,0} - y_{i,1}y_{i,0}\right) + \sum_{i\in W} l_i c_{ri}\, y_i \leq b \quad (6\text{-}3)$$

$$Y_1 \in \{0,1\}^{|F|} \quad (6\text{-}4)$$

$$Y \in \{0,1\}^{|W|} \quad (6\text{-}5)$$

Where $Q(Y_1, Y, \boldsymbol{\xi}(\omega_k))$ is the optimal value of the following problem:

$$\text{Min} \quad \sum_{(ij)\in A} \boldsymbol{q}(\omega_k)_{ij}\, x_{ij} + \sum_{j\in C} g_j u_j \quad (6\text{-}6)$$

$$s.t. \quad \sum_{i\in N} x_{ij} - \sum_{l\in N} x_{jl} = 0 \;; \qquad \forall j \in F \quad (6\text{-}7)$$

$$\sum_{i\in W} x_{ij} + u_j \geq \boldsymbol{d}(\omega_k)_j \;; \qquad \forall j \in C \quad (6\text{-}8)$$

$$\sum_{j\in P} x_{ij} \leq \boldsymbol{s}(\omega_k)_i \;; \qquad \forall i \in S \quad (6\text{-}9)$$

$$r_j\left(\sum_{i\in S} x_{ij}\right) \leq \boldsymbol{m}(\omega_k)_j y_{j,1} \;; \qquad \forall j \in P \quad (6\text{-}10)$$

$$r_j\left(\sum_{i\in P} x_{ij}\right) \leq \boldsymbol{m}(\omega_k)_j (y_{j,1} + y_j) \;; \qquad \forall j \in W \quad (6\text{-}11)$$

$$X \in R_+^{|A|},\; U \in R_+^{|C|} \quad (6\text{-}12)$$

The objective function in Equation (6-1) is to minimize the expected total cost of the supply chain. The first and second terms represent the cost of opening and closing firm-owned facilities, respectively. Please note that if a firm-owned facility has been used previously, there is no cost for opening it at the current time stage. Similarly, if no firm owned facility exists at a node in the previous time stage, then no cost is incurred for closing a facility at that node. The third term is the cost of renting warehouses. The constant l_i (warehouse life time) is multiplied to this term to indicate that a particular warehouse should be rented only when its annual rent multiplied by its lifetime is lower than the cost of building it. The fourth and fifth terms are the initial or base costs required for operating any facility. The sixth term refers to the expected cost of the second-stage problem. Equation (6-2) refers to the constraints saying whereby we can either build a firm-owned warehouse or rent a warehouse at each candidate node. Equation (6-3) refers to the constraint on reconfiguration set by management's predetermined budget limit. Equation (6-4) and (6-5) are binary variable constraints.

Equations (6-6)-(6-12) express the second-stage problem. The objective function of minimizing production/transportation cost and penalty cost is shown in Equation (6-6). Equation (6-7) shows the flow balance constraints at the facility nodes. Equation (6-8) states that the sum of units shipped to and unfilled at a customer node must be greater than or equal to the demand at that node. Constraint in supply quantity at each supplier node is expressed in Equation (6-9). Equations (6-10) and (6-11) are capacity constraints at the plant and warehouse nodes. Specifically, they state that the capacity required to process units of product shipped to any facility node cannot exceed the capacity of that node. Equation (6-12) shows the non-negative variable constraints.

Some issues need to be addressed in order to solve this problem. First, in the case of continuous random parameters, the resulting model obtained from this formulation is an infinitely large two-stage stochastic program. Therefore, in order to solve this problem numerically in practice, we need to approximate this problem into a finite size problem so that we are able to solve it in a tractable manner. Second, due to the nature of the supply chain network design problems, hundreds of random parameters are not uncommon in practice. This high number of random parameters could cause extremely high variance of the objective value. To achieve meaningful interpretation of the problem solution, variance reduction techniques should be used to decrease the variance of the objective function. This study addresses the first issue by using the Sample Average Approximation (SAA) and L-Shaped decomposition methods, and the study addresses the second issue by using some efficient sampling techniques embedded in the SAA algorithm. In particular, we investigate the degree of variance reduction in the objective value of the SRR problem as obtained by using these sampling techniques: Simple Random Sampling (SRS), Latin Hypercube Sampling (LHS), and Sobol' sequence sampling. We discuss this solution approach in the following section.

6.2 Solution approach to the supply chain network redesign under risk problem

To solve the supply chain network redesign problem under consideration efficiently, we use a combination of three methods: the Sample Average Approximation (SAA), L-Shaped decomposition, and efficient sampling methods. Figure 6.1 represents the schematics of the solution procedure to this problem.

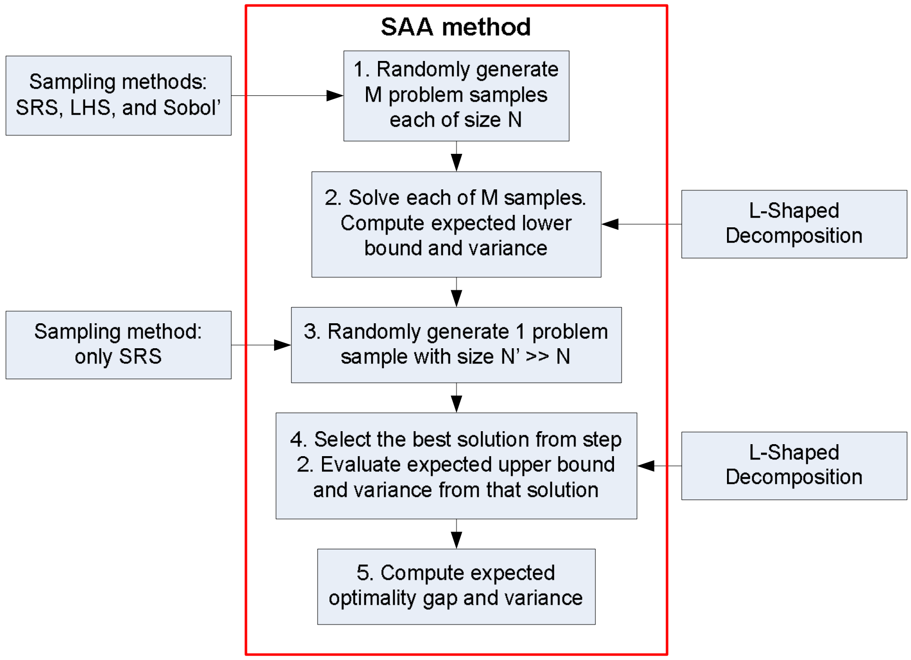

Figure 6-1: Solution procedure to the supply chain network redesign under risk

6.2.1 Sample Average Approximation (SAA)

The SAA algorithm is a backbone of our solution procedure to the supply chain network redesign problem. It is used to approximate the infinitely large problems into the finite and tractable size problems. The steps of the algorithm are shown schematically in Figure 6-1 and are described below which are based on Santoso et al. (2005).

Algorithm 6.1: SAA

Step 1: Randomly generate M i.i.d (independent identically distributed) samples of the SRR problem each of size N, i.e. $(\xi_j^1, ..., \xi_j^N)$ for j = 1, …., M.

Step 2: For each sample j for j =1, 2, …., M, solve the following problem which is derived from Equation (6-1).

$$\text{Min } z = \sum_{i\in F} c_{oi}\left(y_{i,1} - y_{i,1}y_{i,0}\right) + \sum_{i\in F} c_{ci}\left(y_{i,0} - y_{i,1}y_{i,0}\right) + \sum_{i\in W} l_i c_{ri}\, y_i + \sum_{i\in P} c_{bi} y_{i,1}$$

$$+ \sum_{i\in W} c_{bi}(y_{i,1} + y_i) + \frac{1}{N}\sum_{\mathrm{n=1}}^{\mathrm{N}} Q\left(Y_1, Y, \xi_j^n\,(\omega_k)\right)$$

where the notations are the same as in Section 6.1. Let ν_N^j, $\hat{Y}_{1,N}^j$ and $\hat{Y}_N^j$, j =1, …., M be the corresponding optimal objective value and an optimal solution, respectively. Then let $\bar{\nu}_{N,M}$ be a lower statistical bound for the optimal objective value ν^* of the true SRR problem, and $\sigma^2_{\bar{\nu}_{N,M}}$ is an estimate of the variance of this lower bound. Compute

$$\bar{\nu}_{N,M} = \frac{1}{M}\sum_{j=1}^{M} \nu_N^j \qquad \text{and}$$

$$\sigma^2_{\bar{\nu}_{N,M}} = \frac{1}{(M-1)M}\sum_{j=1}^{M}(\nu_N^j - \bar{\nu}_{N,M})^2$$

<u>Step 3:</u> Randomly generate one sample of the SRR problem of size N', where $N' \gg N$.

<u>Step 4:</u> Select the best feasible solution $\hat{Y}_1$ and $\hat{Y}$ of the true problem obtained from step 2 and then evaluate the estimate of $f(\hat{Y}_1, \hat{Y})$ as follows:

$$\tilde{f}_{N'}(\hat{Y}_1, \hat{Y})$$

$$= \sum_{i\in F} c_{oi}\left(\hat{y}_{i,1} - \hat{y}_{i,1}y_{i,0}\right) + \sum_{i\in F} c_{ci}\left(y_{i,0} - \hat{y}_{i,1}y_{i,0}\right) + \sum_{i\in W} l_i c_{ri}\, \hat{y}_i$$

$$+ \sum_{i\in P} c_{bi}\hat{y}_{i,1} + \sum_{i\in W} c_{bi}(\hat{y}_{i,1} + \hat{y}_i) + \frac{1}{N'}\sum_{n=1}^{N'} Q\left(\hat{Y}_1, \hat{Y}, \xi^n(\omega_k)\right)$$

Here ξ^1, …., $\xi^{N'}$ are generated independently of the sample used to obtain $\hat{Y}_1$ and $\hat{Y}$. Then $\tilde{f}_{N'}\left(\hat{Y}_1, \hat{Y}\right)$ will be an estimate of an upper bound on $\nu^*$and its variances can be computed by

$$\sigma^2_{N'}(\hat{Y}_1, \hat{Y}) := \frac{1}{(N'-1)N'}\sum_{n=1}^{N'}(FacCost + Q\left(\hat{Y}_1, \hat{Y}, \xi^n(\omega_k)\right) - \tilde{f}_{N'}\left(\hat{Y}_1, \hat{Y}\right))^2 y$$

where *FacCost* is all cost terms associated with the facilities:

$$FacCost = \sum_{i\in F} c_{oi}\left(\hat{y}_{i,1} - \hat{y}_{i,1}y_{i,0}\right) + \sum_{i\in F} c_{ci}\left(y_{i,0} - \hat{y}_{i,1}y_{i,0}\right) + \sum_{i\in W} l_i c_{ri}\,\hat{y}_i + \sum_{i\in P} c_{bi}\hat{y}_{i,1} + \sum_{i\in W} c_{bi}(\hat{y}_{i,1} + \hat{y}_i)$$

Step 5: Compute the expected optimality gap between the lower and upper bounds:

$$gap_{N,M,N'}(\hat{Y}_1, \hat{Y}) = \tilde{f}_{N'}(\hat{Y}_1, \hat{Y}) - \bar{v}_{N,M}$$

And the estimated variance of the gap is

$$\sigma^2_{gap} = \sigma^2_{N'}(\hat{Y}_1, \hat{Y}) + \sigma^2_{\bar{v}_{N,M}}$$

It is our intention to provide readers with only a brief description of the SAA algorithm. The steps given above are specific to solving our supply chain network redesign only. For more general results and its derivations of upper bound, lower bound, variance and convergence rate, the readers are referred to the works in (Kleywegt et al., 2001) and (Linderoth et al., 2006).

6.2.2 L-Shaped decomposition method

In this research, the L-Shaped decomposition method is used to solve the mixed-integer programs generated in steps 2 and 4 of the SAA method, as shown in Figure 6-1. The L-Shaped decomposition method is essentially equivalent to the well-known Benders decomposition method. The only difference is the point at which the optimality cuts are added into the master (first stage) problem. Since the Benders decomposition method is originally designed for solving a deterministic problem, it adds an optimality cut based on only a single scenario (because the problem is deterministic and has only one scenario). On the other hand, the L-Shaped decomposition method consists of multiple scenarios in the second stage problem; therefore, an optimality cut is generated based on multiple scenarios. That is, coefficients of an optimality cut are generated from the expectation across all scenarios under consideration. A complete version of the L-Shaped decomposition method can be found in (Birge and Louveaux, 2000) and (Van Slyke, 1969). The following is the specialized version of the L-Shaped decomposition method used to solve problems in supply chain network redesign applications, i.e., Equations (6-1)-(6-12) only. The notations used in this algorithm are similar to those in Section 6.1, unless otherwise specified.

Algorithm 6.2: L-Shaped decomposition

Initial step: Set lower bound (lb) = $-\infty$ and upper bound (ub) = $+\infty$. Set o, v = 0 where o is the number of optimality cuts and v is the iteration number

Step 1: Set v = v+1 and solve the master problem Equation (6-13)-(6-18):

$$\text{Min } z = \sum_{i\in F} c_{oi}\left(y_{i,1} - y_{i,1}y_{i,0}\right) + \sum_{i\in F} c_{ci}\left(y_{i,0} - y_{i,1}y_{i,0}\right) + \sum_{i\in W} l_i c_{ri}\, y_i + \sum_{i\in P} c_{bi} y_{i,1} + \sum_{i\in W} c_{bi}(y_{i,1} + y_i) + \theta \tag{6-13}$$

$$s.t. \quad y_{i,1} + y_i \le 1 \qquad \forall i \in W \tag{6-14}$$

$$\sum_{i\in F} c_{oi}\left(y_{i,1} - y_{i,1}y_{i,0}\right) + \sum_{i\in F} c_{ci}\left(y_{i,0} - y_{i,1}y_{i,0}\right) + \sum_{i\in W} l_i c_{ri}\, y_i \le b \tag{6-15}$$

$$\sum_{i\in F} E_{1,t,i} y_{i,1} + \sum_{i\in W} E_{t,i} y_i + \theta \ge e_t \qquad t = 1, 2, \dots, o \tag{6-16}$$

$$\theta \in R \tag{6-17}$$

$$Y_1 \in \{0,1\}^{|F|},\ Y \in \{0,1\}^{|W|} \tag{6-18}$$

where Equation (6-16) is the optimality cut. $E_{1,t,i}$ and $E_{t,i}$ are the coefficient of the optimality cut obtained from step 3, and e_t is the constant obtained from step 3 as well. If o equals zero, then let $\theta = -\infty$. If the problem is infeasible, terminate the algorithm because the original problem is infeasible. Otherwise, let Y_1^v, $Y^{v,}$ and θ^v be an optimal solution to the master problem, lb = z, and go to step 2.

<u>Step 2:</u> Let k be an index of scenarios and N is the number of scenarios generated in step 1 of the SAA algorithm. For k = 1, 2, …., N, solve the following subproblem Equations (6-19)-(6-25).

$$\text{Min} \quad \sum_{(ij)\in A} q_{k,ij}\, x_{ij} + \sum_{j\in C} g_j u_j \tag{6-19}$$

$$s.t. \quad \sum_{i\in N} x_{ij} - \sum_{l\in N} x_{jl} = 0 \;; \qquad \forall j \in F \tag{6-20}$$

$$\sum_{i\in W} x_{ij} + u_j \geq d_{k,j} \;; \qquad \forall j \in C \tag{6-21}$$

$$\sum_{j\in P} x_{ij} \leq s_{k,i} \;; \qquad \forall i \in S \tag{6-22}$$

$$r_j\left(\sum_{i\in S} x_{ij}\right) \leq m_{k,j} y_{j,1}^{v} \;; \qquad \forall j \in P \tag{6-23}$$

$$r_j\left(\sum_{i\in P} x_{ij}\right) \leq m_{k,j}(y_{j,1}^{v} + y_j^{v}) \;; \qquad \forall j \in W \tag{6-24}$$

$$X, U \in R_+^{|A|} \tag{6-25}$$

<u>Step 3:</u> Let $\alpha_k^v, \beta_k^v, \gamma_k^v, \eta_k^v, and\ \mu_k^v$ be vectors of optimal dual solutions associated with Equations (6-20), (6-21), (6-22), (6-23), and (6-24) in scenario k at iteration v. Also, let $E_{1,o+1}$ be a vector of cut coefficients associated with primal variables $y_{i,1}$ in Equation (6-16) and E_{o+1} be a vector of cut coefficients for primal variables y_i in Equation (6-16) as well. Define:

$$E_{1,o+1,j} = \sum_{k=1}^{N} p_k\, \eta_{k,j}^{v}(-m_{k,j}) \qquad \forall j \in P \tag{6-26}$$

$$E_{1,o+1,j} = \sum_{k=1}^{N} p_k\, \mu_{k,j}^{v}(-m_{k,j}) \qquad \forall j \in W \tag{6-27}$$

Let $E_{1,o+1}$ be a vector consisting of elements $E_{1,o+1,j}$ from Equation (6-26) followed by elements $E_{1,o+1,j}$ from Equation (6-27). Then define

$$E_{o+1,j} = \sum_{k=1}^{N} p_k\, \mu_{k,j}^{v}(-m_{k,j}) \qquad \forall j \in W \tag{6-28}$$

$$e_{o+1} = \sum_{k=1}^{N} p_k \left(\sum_{j=1}^{|C|} \beta_{k,j}^{v} d_{k,j} + \sum_{i=1}^{|S|} \gamma_{k,i}^{v} s_{k,i}\right) \tag{6-29}$$

where e_{o+1} is the right-hand side in Equation (6-16) for the $(o+1)^{th}$ cut and p_k is probability that scenario k will occur. In our case, it is assumed that all scenarios have equal probability of occurrence, therefore, $p_k = 1/N$.

Step 4: Let

$$w^v = e_{o+1} - (E_{1,o+1}Y_1^v + E_{o+1}Y^v).$$

If $(w^v - \theta^v)/lb \leq \delta$, where δ is pre-specified tolerance in percent, terminate the algorithm. Y_1^v and Y^v is the optimal solution to the problem. The upper bound, and hence the optimal objective value, is $ub = (lb - \theta^v) + w^v$. Otherwise, let o = o+1, and then return to step 1.

6.2.3 Efficient sampling techniques

To compare the degree of variance reduction and the computational efficiency of each sampling technique for problems of different sizes (or numbers of random parameters), three different sampling strategies are applied in this work, i.e., Simple Random Sampling (SRS), Latin Hypercube Sampling (LHS), and Sobol' sequence sampling. SRS is a simple method that randomly generates uniform points within the range [0,1) and commonly used in any Monte Carlo Simulation model. It is expected that readers of this work are all familiar with this sampling strategy. Therefore, in this section, we only explain topics related to the LHS and Sobol' sampling. If required, discussions of the SRS method can be found in several sources, among which are (Lemieux, 2009) and (Gentle, 2004).

6.2.3.1 Latin Hypercube Sampling (LHS)

The Latin Hypercube Sampling (LHS) method was first proposed in (McKay et al., 1979). It is a type of stratified sampling techniques which basically partitions a sample space of a random parameter into several intervals and then samples one point from each interval. Hence, sufficient uniformity in a single dimension or variable is ensured. Given that the number of sampled points from each technique are equal, we would expect this strategy to yield a point set with a higher degree of uniformity than that yielded by SRS. In the original version of LHS, the sampled point in each region is randomly selected. However, to obtain perfect uniformity in a single dimension, one can deterministically select a center of the interval as a sampled point if the sample space is partitioned into intervals of equal length. This deterministic approach, though it increases the dependency of the sampling points, will not affect the properties of the SAA method. As we can see in step 1 of the SAA algorithm (Section 6.2.1), the i.i.d. property of the sampling points is required only between samples of the SRR problem. That is, the independent sampling points are not necessary within each problem sample in order to correctly compute the expected variance. Therefore, this deterministic sampling strategy is well-suited to this case.

For multiple random parameters, a similar approach can be taken for obtaining a set of uniform points for each parameter, and then we randomly permute the order of sampled points in each parameter. Lastly, we match sampled points from all the permuted arrays to form a multi-dimensional random parameter. The steps for generating a point set for the multi-dimensional problem in LHS are given in detail next.

Algorithm 6.3: LHS

Step 1: Let n = the index of samples, N = the total number of samples, k = the index of random parameters, and K = the total number of random parameters.

Step 2: For k = 1, 2, …., K and n = 1, 2, …., N, let

$$x_{n,k} = F_k^{-1}\left(\frac{n-0.5}{N}\right)$$

where F_k is a cumulative distribution function of random parameter k.

Step 3: For k = 1, 2, …., K, randomly permute the array containing the sampled points associated with parameter x_k.

Step 4: Based on the new order of the permuted arrays, match the points from all the random parameters to obtain N values of the K dimensional sampling point.

6.2.3.2 Sobol' sequence sampling

Sobol' sequence sampling, proposed in (Sobol', 1967), is a type of low-discrepancy sampling techniques that are used in the Quasi-Monte Carlo method. It can be considered a multi-dimensional uniform sampling technique since it aims to place n points on to a k-dimensional hypercube as uniformly as possible, where n, k > 1. In this subsection, we first discuss an algorithm for constructing a Sobol' sequence for a single-dimensional problem. An algorithm for generating a multi-dimensional problem can be extended from the construction of the single-dimensional case. To illustrate the steps in the algorithms, an example of a single-dimensional and an example of a multi-dimensional Sobol' sequence construction will be given at the end of each algorithm.

Algorithm 6.4: Sobol' sequence with a single random parameter (summarized from (Lemieux, 2009))

Step 1: For a nonnegative integer i, define

$$i = \sum_{l=0}^{\infty} a_l(i) b^l$$

where we assume infinitely many coefficients $a_l(i)$ are zero and $a_l(i) < b$. For the Sobol' sequence, b is set to 2. However, for the other low-discrepancy sequences, b may be set to different values.

Step 2: Define the radical-inverse function in base b, denoted by ϕ_b, as

$$\phi_b(i) = \sum_{l=0}^{\infty} a_l(i) b^{-l-1}.$$

Hence, $\phi_b \in [0,1)$. Then the ith term of the Sobol' sequence, denoted by u_i, is given by $\phi_b(i-1)$ for $i \geq 1$.

Example: single random parameter case

Let us assume that we need to generate the 1st to the 3rd terms of the Sobol' sequence. Therefore, we need to find $\phi_2(0)$, $\phi_2(1)$, and $\phi_2(2)$.

Step 1: Write 0, 1, and 2 in the expansion terms as shown in Step 1. Therefore, we obtain $0 = 0 \times 2^0$, $1 = 1 \times 2^0$, and $2 = 0 \times 2^0 + 1 \times 2^1$.

Step 2: Therefore, we obtain $u_1 = \phi_2(0) = 0 \times 2^{-0-1} = 0$, $u_2 = \phi_2(1) = 1 \times 2^{-0-1} = 1/2$, $u_3 = \phi_2(2) = 0 \times 2^{-0-1} + 1 \times 2^{-1-1} = {}^1\!/_4$.

As we can notice from the algorithm above, if we simply use this algorithm to generate random points in multiple dimensions, we will get exactly the same sequence for every random parameter. Therefore, in order to extend the algorithm to the multi-dimensional problem, the coefficient $a_l(i)$ must be transformed before inputting into the radical-inverse function in step 2 in order to have differences among the sequences. This method is called a linear transformation. The algorithm for a multi-dimensional Sobol' sequence with a linear transformation method is given next.

Algorithm 6.5: Sobol' sequence with multiple random parameters (summarized from (Lemieux, 2009)

Step 1: For a nonnegative integer i, define

$$i = \sum_{l=0}^{\infty} a_l(i) b^l$$

Where we assume infinitely many coefficients $a_l(i)$ are zero and $a_l(i) < b$. Set b = 2.

Step 2: Define the vector

$$(\tilde{a}_{j,0}(i), \tilde{a}_{j,1}(i), \tilde{a}_{j,2}(i), \ldots.)^T = C_j \cdot (a_0(i), a_1(i), a_2(i), \ldots.)^T$$

for j = 1, 2, …., k, where k is the number of random parameters. C_j is the generating matrix of each parameter and can be obtained according to step 3 to Step 6 as follows.

Step 3: For each coordinate j (each random parameter), we first define a primitive polynomial, denoted by $p_j(z)$, in F_2, where F_m denotes Galois field with m elements:

$$p_j(z) = z^{d_j} + a_{j,1}z^{d_j-1} + \cdots + a_{j,d_j}$$

where each $a_{j,l} \in \mathrm{F}_2$ and d_j is the degree of $p_j(z)$.

Step 4: Define d_j direction numbers of the form

$$v_{j,r} = \frac{m_{j,r}}{2^r}$$

where $m_{j,r}$ is an odd integer between 1 and 2^r-1 for r = 1, 2, …., d_j. Then write $v_{j,r}$ in the binary expansion form:

$$v_{j,r} = v_{j,r,1}2^{-1} + v_{j,r,2}2^{-2} + \cdots + v_{j,r,d_j}2^{-d_j}$$

Step 5: Determine the number of columns required in each C_j. This number depends on how many sampling points we need to generate. If we need to generate n = b^k points, there must be k columns in C_j. For instance, if we need to generate 16 points, then 16 = 2^4. So, there are 4 columns in C_j.

Then, we may need to define additional direction numbers. If this is the case, the number of additional direction numbers is k - d_j. These additional direction numbers are defined as

$$v_{j,r} = a_{j,1}v_{j,r-1} \oplus \ldots \oplus a_{j,d_j-1}v_{j,r-d_j+1} \oplus v_{j,r-d_j} \oplus \left(v_{j,r-d_j}/2^{d_j}\right),$$

where $\oplus$ represents the exclusive-or operation on binary vectors.

Step 6: The r^{th} column of C_j is then formed by the expansion of $v_{j,r}$. That is, each direction number is assigned to a column of C_j and fills the column with its binary representation.

Step 7: The j^{th} coordinate of the i^{th} point of the Sobol' sequence is defined as

$$u_{ij} = \phi_2(i-1) = \textstyle\sum_{l=0}^{\infty} \tilde{a}_{j,l}(i-1)b^{-l-1}.$$

The next example illustrates how the algorithm for the multi-dimensional Sobol' sequence can be applied to a simple case.

Example: multiple random parameter case (adapted from (Lemieux, 2009))

Let us assume that we are generating a Sobol' sequence for the random variable j = 3 and that we need to generate 8 sampling points. Therefore, we need to find $\phi_2(0)$, $\phi_2(1)$, $\phi_2(2)$, …., $\phi_2(7)$.

Step 1: Write 0, 1, 2, …., 7 in the expansion terms. Therefore, we obtain $0 = 0 \times 2^0$, $1 = 1 \times 2^0$, $2 = 0 \times 2^0 + 1 \times 2^1$, $3 = 1 \times 2^0 + 1 \times 2^1$, $4 = 0 \times 2^0 + 0 \times 2^1 + 1 \times 2^2$, $5 = 1 \times 2^0 + 0 \times 2^1 + 1 \times 2^2$, $6 = 0 \times 2^0 + 1 \times 2^1 + 1 \times 2^2$, $7 = 1 \times 2^0 + 1 \times 2^1 + 1 \times 2^2$.

Step 2: Then we need to find

$(\tilde{a}_{3,0}(i), \tilde{a}_{3,1}(i), \tilde{a}_{3,2}(i))^T = C_3 \cdot (a_0(i), a_1(i), a_2(i))^T$, for i = 0, 1, …., 7

where C_3 can be constructed from Step 3 to Step 6 as follows.

Step 3: Take $p_3(z) = z^2 + z + 1$.

Step 4: Since the degree of $p_3(z)$ is two, we need to generate two direction numbers:

$$v_{3,1} = \frac{m_{3,1}}{2^1} = \frac{1}{2}$$

$$v_{3,2} = \frac{m_{3,2}}{2^2} = \frac{3}{4}$$

In this case, $m_{3,2}$ is arbitrarily chosen to 3. Then write $v_{3,1}$ and $v_{3,2}$ in the binary expansion form:

$$v_{3,1} = \frac{1}{2} = v_{3,1,1}2^{-1} + v_{3,1,2}2^{-2} = (1)2^{-1} + (0)2^{-2}$$

$$v_{3,2} = \frac{3}{4} = v_{3,2,1}2^{-1} + v_{3,2,2}2^{-2} = (1)2^{-1} + (1)2^{-2}$$

So, in vector representation, $v_{3,1} = (1,0)$ and $v_{3,2} = (1,1)$.

Step 5: Since we need 8 sampling points, $8 = 2^3$; hence C_3 requires 3 columns. So we need one more direction number:

$$v_{3,3} = a_{3,1}v_{3,2} \oplus v_{3,1} \oplus \left(v_{3,1}/2^2\right)$$

where $v_{3,1}/2^2 = \frac{1}{2\times 2^2} = \frac{1}{8}$. Then $\frac{1}{8} = v_{3,3,1}2^{-1} + v_{3,3,2}2^{-2}v_{3,3,3}2^{-2} = (0)2^{-1} + (0)2^{-2} + (1)2^{-3}$. So, in vector representation, $v_{3,1}/2^2 = (0,0,1)$. We already know from step 3 that $a_{3,1} = 1$. Then we get

$$v_{3,3} = \begin{pmatrix}1\\1\end{pmatrix} \oplus \begin{pmatrix}1\\0\end{pmatrix} \oplus \begin{pmatrix}0\\0\\1\end{pmatrix} = \begin{pmatrix}0\\1\\1\end{pmatrix}.$$

<u>Step 6:</u> Therefore, C_3 is defined as

$$C_3 = \begin{pmatrix}1 & 1 & 0\\0 & 1 & 1\\0 & 0 & 1\end{pmatrix}.$$

<u>Step 7:</u> We obtain

$$(\tilde{a}_{3,0}(i), \tilde{a}_{3,1}(i), \tilde{a}_{3,2}(i))^T = \begin{pmatrix}1 & 1 & 0\\0 & 1 & 1\\0 & 0 & 1\end{pmatrix} \cdot (a_0(i), a_1(i), a_2(i))^T, \quad \text{for i = 0, 1, …., 7}$$

where $(a_0(i), a_1(i), a_2(i))^T$ can be obtained from step 1. For example, for i = 1,

$$(\tilde{a}_{3,0}(1), \tilde{a}_{3,1}(1), \tilde{a}_{3,2}(1))^T = \begin{pmatrix}1 & 1 & 0\\0 & 1 & 1\\0 & 0 & 1\end{pmatrix} \cdot (1,0,0)^T = (1,0,0)^T$$

Therefore, we obtain

$$u_{23} = \phi_2(1) = \sum_{l=0}^{\infty} \tilde{a}_{j,l}(1)2^{-l-1} = (1)2^{-1} + (0)2^{-2} + (0)2^{-3} = 1/2.$$

The example above is given for the case where the index of the random parameter j is 3. A similar procedure can be used to generate the Sobol' sequences for other values of j, e.g., 1, 2, 4, 5, and so on. Furthermore, other than the multi-dimensional Sobol' sequence algorithm described above, we also apply the Gray code technique to the real implementation of this sampling technique in this work to accelerate the computational time of the Sobol' sequence algorithm. The detail of the Gray code implementation can be found in (Antonov and Saleev, 1980).

As one may notice, an obvious issue when using the Sobol' sequence sampling technique within the SAA is that, in step 1 of the SAA algorithm, we need to generate M problem samples each of size N, and it is required that the M samples must be i.i.d. If we simply use the multi-dimensional Sobol' sequence given above in the SAA method, all the problem samples will have the same data points and hence the requirement of i.i.d may not be satisfied. To address this issue, the random digital shift can be used to develop the i.i.d property for these M samples.

The random digital shift is one of the randomization techniques used in the randomized Quasi-Monte Carlo method. The basic idea of this technique is to randomly shift a point set in each problem sample such that (1) the uniformity within each problem sample is still maintained, and (2) the shifted point sets of the M problem samples are i.i.d. (independent identically distributed) and follow $U([0,1)^k)$, where k is the number of random parameters or the problem dimension. As a result, after applying this random digital shift technique, the variance of the lower bound of the objective value in step 2 of the SAA method can be computed appropriately. Therefore, in the present study, we also implement the random digital shift technique when using the Sobol' sequence sampling technique in the SAA method. For more detail about the random digital shift technique, readers are referred to (Lemieux, 2009).

So far, in this section, we have discussed the integrated solution approach to our supply chain network redesign problem stated in Section 6.1. In the next section, in order to observe and compare the efficiency of the three sampling techniques described earlier, i.e., SRS, LHS, and Sobol' sequence, numerical experiments are conducted to solve the network redesign problem under consideration here and some of the performance measures are observed to determine the most efficient sampling technique for different problem sizes.

6.3 Numerical experiments and results

6.3.1 Problem settings

In order to determine the efficiency of the three sampling techniques, example problems were arbitrarily created from the SRR problem from Section 6.1 and numerical experiments were conducted in order to solve the problems using the solution approach discussed earlier. Three observations were recorded including the variance of the optimality gap, total computation time, and efficiency of the sampling techniques. The factors and parameters of the experiments are given below.

- Sampling techniques used: SRS, LHS, and Sobol' sequence.
- Problem sizes: small, medium, and large. Table 6-1 shows the number of nodes and random parameters in each problem size.

Table 6-1: Number of nodes and random parameters

Problem size	**Small**	**Medium**	**Large**
Total No. of nodes	**10**	**20**	**40**
No. of suppliers	3	6	12
No. of plants	1	2	4
No. of warehouses	2	4	8
No. of customers	4	8	16
No. of random parameters	**23**	**72**	**248**

- Parameters in the SAA method: for all experiments, M=20, N=40, and N'=1000.
- For the required capacity for producing a unit of product (r_j), we assume that $r_j = 1$ for all $j \in F$. The penalty cost of the product shortage at customer nodes equals 100 per unit.
- Network redesign parameters: all random parameters, that is, the shipping costs, customer demands, supply quantities, and facility capacities, were assumed to be continuous and normally distributed with standard deviations equal to 10% of the mean values. Tables 6-

2 to 6-6 show the data, including the parameter mean values and fixed costs, of the network redesign for the small size problem, i.e. total number of node is 10. Details regarding the medium- and large-size problems can be found in Appendix E.

Table 6-2: Parameters and fixed costs for facilities for the small problem

Fixed cost	**Plant**	**Warehouse**	
	no.1	**no.1**	**no.2**
Previously owned facility	no	no	no
Open cost	3000000	1000000	900000
Rent cost	-	105000	85000
Close cost	450000	150000	135000
Base operating cost	540000	180000	162000
Reconfiguration budget	99999999		

Table 6-3: Mean shipping costs for the small problem

Shipping costs (s=supplier, p=plant, w=warehouse, c=customer)								
s	**p**	**cost/unit**	**p**	**w**	**cost/unit**	**w**	**c**	**cost/unit**
1	1	5	1	1	7	1	1	10
2	1	4		2	5		2	9
3	1	3					3	8
							4	9
						2	1	8
							2	7
							3	10
							4	10

Table 6-4: Mean customer demands for the small problem

Demand	**Units**
1	15000
2	13000
3	20000
4	17000

Table 6-5: Mean supply quantities for the small problem

Supply	**Units**
1	50000
2	43000
3	57000

Table 6-6: Mean facility capacities for the small problem

Facility capacity			
plant	**Cost**	**warehouse**	**Cost**
1	100000	1	50000
		2	40000

- Simulation runs: for each experiment, the simulation was run for 10 replications to compute the average of each observation.

In order to compare the efficiency of the three sampling techniques, three performance measures were observed, which are:

- Variance of the optimality gap: in Step 5 of the SAA method, the optimality gap and its variance are computed. It is necessary for this variance to be as low as possible so that accurate information about the optimality gap can be obtained.
- Total computational time: the total computational time of the solution was also recorded for each experiment. To precisely compare the total computational times, we observed real CPU times instead of wall clock times.
- Efficiency: to identify the best sampling technique for each problem size, we need to consider both variance and computational time simultaneously. However, usually variance and computational time are measures that must be traded off; for example, sampling techniques providing low variance usually require a longer time to generate a point set. Therefore, in this case, we judged the performance of the sampling techniques based on their efficiency and it is defined as:

$$Eff_i = \frac{Var_{min}}{Var_i} \times \frac{Time_{min}}{Time_i}, \qquad \text{i = SRS, LHS, and Sobol'}$$

where

Eff_i = efficiency of sampling technique i

Var_{min} = minimum optimality gap variance obtained among the three techniques

Var_i = optimality gap variance obtained from sampling technique i

$Time_{min}$ = minimum total computational time used among the three techniques

$Time_i$ = total computational time used by sampling technique i

We applied the three different sampling techniques to the three problem sizes, such that 9 experiments were conducted. The numerical simulation was performed on a server-grade workstation at the High Performance Computing (HPC) group at the Penn State University. Specifically, the programming code was implemented using ILOG Concert technology for JAVA with CPLEX 11.0 and run on a Linux machine with Quad 2.6 GHz AMD Opteron, 32 GB of RAM. The results of these experiments are presented and discussed next.

6.3.2 Results and discussion

First we observed the result of the optimality gap variance for each sampling technique. Table 6-7 and Figure 6-2 show the results of this observation.

Table 6-7: Average of optimality gap variance

	No. of random parameters		
Method	**23**	**72**	**248**
SRS	2.154E+07	1.779E+08	5.429E+09
LHS	8.859E+06	6.682E+07	4.383E+09
Sobol'	1.187E+07	7.371E+07	4.366E+09

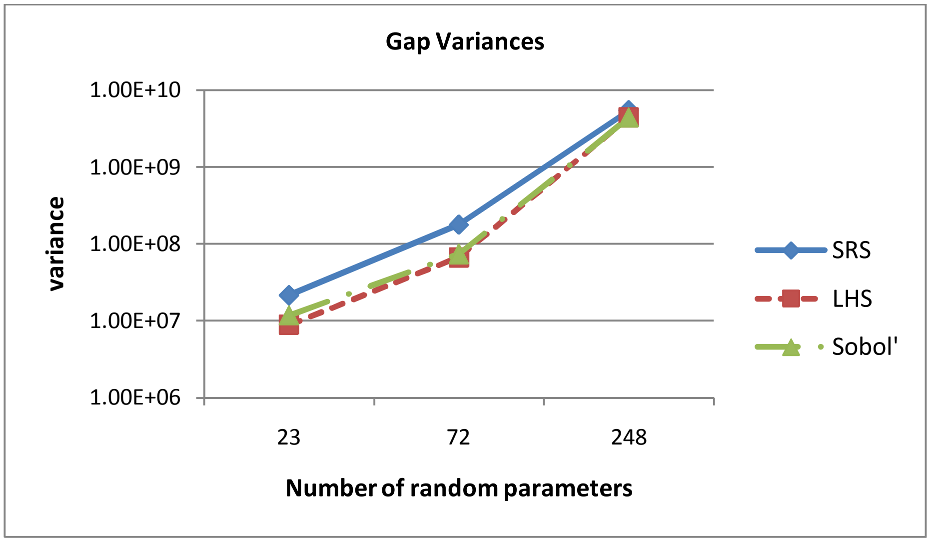

Figure 6-2: Results of the average optimality gap variances

As we can see from the table and plot, the SRS technique generally results in a larger variance than LHS and the Sobol' sequence do in general; which confirms that both LHS and Sobol' sequence can significantly reduce the variance of the optimality gap in the supply chain network redesign problem. For the small problem size (23 random parameters), LHS tends to perform better than the Sobol' sequence. In particular, LHS yields an average variance of approximately 8.859×10^6 approximately, which is lower than the average variance of approximately 1.187×10^7 that the Sobol' sequence produces. However, as the number of the random parameters increases, the variances from the LHS and the Sobol' sequence are getting closer. As we can see in the cases of medium and large problem sizes, the variances from the LHS and the Sobol' sequences are very close. Specifically, for the large problem size, the variance from the LHS is slightly larger than, but still comparable to, the variance from the Sobol' sequence. One explanation for this result could be that the LHS technique is designed to generate good uniformity in a single dimension. As we can see from algorithm 6.3, if a point is sampled at the center of each interval, we could get perfect uniformity in a single dimension. However, once the number of dimensions or random parameters increases, the uniformity of a point set may be distorted. Therefore, LHS tends to perform worse in cases with higher dimensions as we can see in the medium and large problem sizes that the variances from the LHS and Sobol' sequence are really close compared to each other.

The next point of interest is the total computational time. As Table 6-8 and Figure 6-3 show, the computational times of the three sampling techniques are considered comparable. However, for large problem size, it is noticed that the LHS technique required a slightly longer time (the difference is insignificant) in order to generate the sampling points and solve the problem, that is, the LHS-based SAA required approximately 463 seconds to obtain the solution which is higher than 438 and 435 seconds from the SRS- and Sobol' sequence-based SAA respectively.

Table 6-8: Average total computational time (seconds)

	No. of random parameters		
Method	**23**	**72**	**248**
SRS	4.753	21.951	438.281
LHS	4.887	24.572	462.526
Sobol'	4.754	24.063	435.109

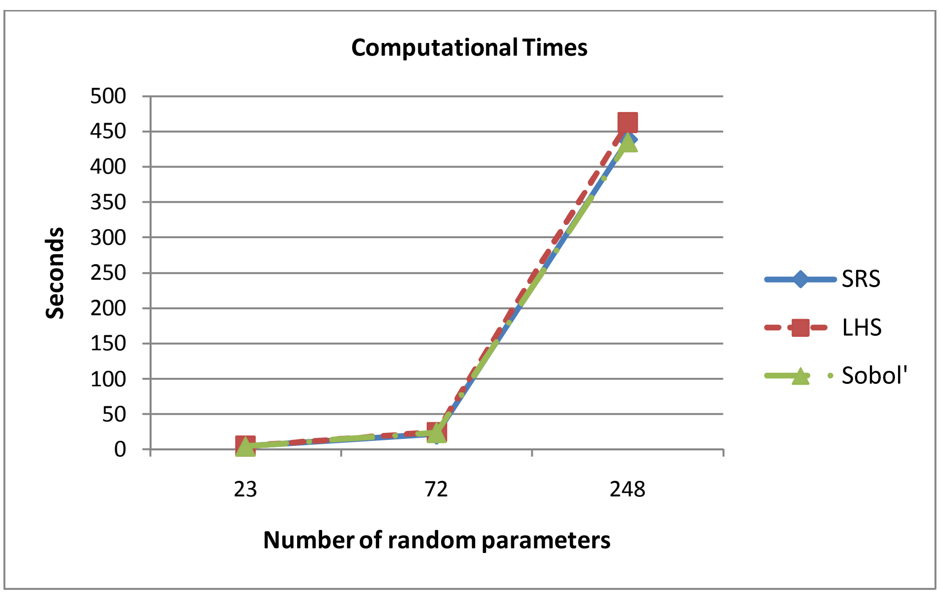

Figure 6-3: Results of the average total computational times

Finally, the efficiency of the sampling techniques was computed for each problem size in order to determine the most appropriate sampling method. According to Table 6-9 and Figure 6-4, LHS and the Sobol' sequences provide us with better performance than the SRS for all problem sizes. Particularly, for the small problem size, LHS is more efficient than the Sobol' sequence. However, the efficiency of the Sobol' sequence improves as the size of the problem increases. For example, the efficiency of the Sobol' sequence is 0.827, which is comparable to the LHS's efficiency of 0.893. In addition, the result shows that the Sobol' sequence performs even more efficient than LHS when the problem size is large, i.e., 1.000 for the Sobol' sequence and 0.937 for the LHS. As a result, it can be concluded that the LHS technique should be preferred for solving a supply chain network redesign problem with a small size; however, when the problem size is large, the Sobol' sequence technique is a better approach.

Table 6-9: Efficiency of the sampling techniques

	No. of random parameters		
Method	**23**	**72**	**248**
SRS	0.411	0.376	0.798
LHS	0.973	0.893	0.937
Sobol'	0.746	0.827	1.000

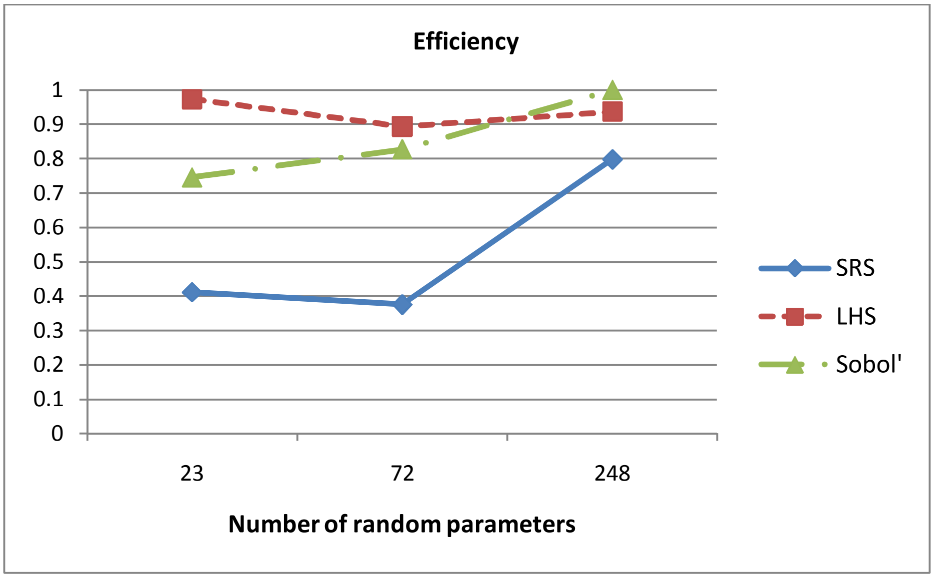

Table 6-10: Results for efficiency

In summary, this chapter discussed the components of the stochastic network optimization module in the network reconfiguration system. The new configuration decision is executed by the two-stage stochastic programming model representing the problem of redesigning a supply chain network under uncertainty problem. Two issues associated with this problem formulation were considered, i.e., an infinitely or extremely large problem size and an exceedingly high variance of the optimality gap. We successfully implemented the Sample Average Approximation (SAA) and L-Shaped decomposition methods in order to address the issue of problem size: the SAA method is capable of approximating the infinite-size problem to a finite size; hence, it can be solved numerically in a finite time, and the L-Shaped decomposition method can help to accelerate the computation time for each mixed-integer program randomly generated in the SAA method. For the second issue, the optimality gap variance, we applied three different sampling strategies to determine the best technique in terms of variance reduction and computational time. The three sampling techniques include the Simple Random Sampling (SRS), the Latin Hypercube Sampling (LHS), and the Sobol' sequence techniques.

According to the experimental result, we can see that, for the problem with the low number of random parameters, the LHS outperformed the other two techniques in terms of the variance reduction. However, for the medium and high numbers of random parameters, the variances obtained from LHS and the Sobol' sequences are comparable. In addition, the computational times for solving the network redesign problems were also observed. As shown in the numerical result, the computational times of all the three sampling techniques are fairly comparable and may not be distinguishable in practice. And, to determine the best sampling strategy for each problem size, the efficiency of each sampling strategy was computed: It is revealed from the result that LHS technique is recommended for solving the small-size supply chain network redesign problem; however, when the problem is large, the Sobol' sequence technique might be a more efficient approach.

Once the new supply chain network configuration has been redesigned by the stochastic network optimization module, it will be sent to the network configuration controller in the system. Then the network configuration controller restructures the supply chain network according to this new configuration. In this research, this reconfiguration task is performed using the Service Oriented Architecture (SOA) for a supply chain. The architecture and implementation of this SOA-based reconfigurable supply chain are the main topics of the next chapter.

Chapter 7

Service-Oriented Architecture for a Supply Chain

The supply chain network configuration obtained from Equations (6-1)-(6-12) is capable of fortifying a supply chain subject to only one type of risk. However, if a different risk type were to be introduced, the network should reconfigure again depending on the new risk type. Such a highly adaptive and dynamic supply chain requires agility in its information system architecture in order to allow all participating and candidate nodes to work together to deliver product to customers. In this chapter, we discuss the development of the last component, i.e., the network configuration controller, and the architecture design of a reconfigurable supply chain network based on Service-Oriented Architecture (SOA) in practice. We will begin by stating some of the requirements of a reconfigurable supply chain network. Then we discuss the main functions of the network configuration controller in the reconfigurable supply chain. Finally, we propose an architecture design of a reconfigurable supply chain network according to its required properties.

7.1 Requirements of a reconfigurable supply chain network

The main purpose of a reconfigurable supply chain network is to reconfigure or adjust its configurations once a risk event is anticipated. The required properties of a reconfigurable supply chain network are as follows:

- Ability to operate in a dynamic risk environment: it is assumed that only one risk event occurs at any given time in the considered supply chain. However, different risk types may replace the previous risk type over time.

- Autonomous nodes in the supply chain: each node in the supply chain should have its own reasoning process because each node represents a single company in the supply chain.
- Scalability of the supply chain network: over time, the supply chain network may be expanded or shrunk depending on the economic situation. Therefore, it is important that the supply chain should be able to add more nodes to or remove some nodes from its network. In particular, the communication infrastructure should be able to handle variation in information exchanges due to such changes.
- Discoverability: from time to time, new nodes may enter into or previous nodes may leave the environment dynamically. It is important that a company is able to update a list of available nodes in its environment in order to form and maintain the most robust supply chain possible. This can be made possible by discoverability.
- Interoperability among diverse IT platforms: different nodes may operate on different IT platforms. Interoperability is the ability that two nodes can be properly connected even though their enterprise application platforms are different. To achieve a reconfigurable supply chain that frequently changes its trading partners, this property is one of the most important characteristics of the network.
- Loose coupling between nodes: loose coupling or low dependency between nodes is a situation in which a node requires only awareness of the existence of its trading partners. In other words, the node needs not to know a complete set of communication protocols in order to send and receive messages from its partners. The only thing the node needs to know is the physical addresses of its partners on the Internet network. Once the node knows its partners' addresses, it can contact and communicate with the partners via a standardized messaging method.

In the next section, we will discuss an architecture design for the information system that has the required properties as stated. As will be discussed, SOA is an architecture that enables the design of a reconfigurable supply chain network under risk.

7.2 Architecture design of an SOA-based reconfigurable supply chain

Let us consider again the overview of the reconfigurable supply chain network and its procedure, as illustrated in Figure 4-1 and Figure 4-2, respectively. The details of the development of the uncertain parameter database and the stochastic network optimization in Figure 4-1 and for steps 1 to 3 in Figure 4-2 are outlined in Chapters 5 and 6. In this section, we will focus on the functions and development of the last component of the reconfigurable supply chain network, namely, the network configuration controller. The procedure of the network reconfiguration previously shown in Chapter 4 is repeated in Figure 7-1 for readers' convenience.

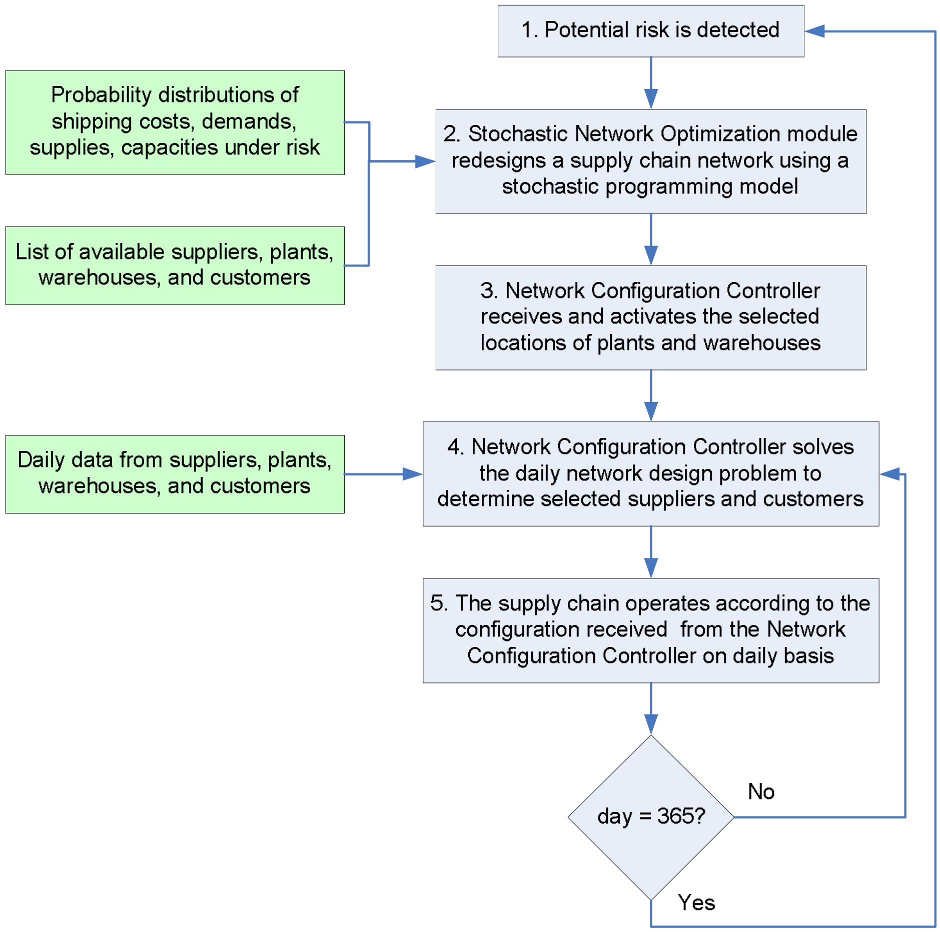

Figure 7-1: Procedure for network configuration (repeated)

7.2.1 Network configuration controller

In step 4 of Figure 7-1, the daily supply chain network configuration must be determined by the network configuration controller. After receiving the locations of the selected plants and warehouses from the stochastic network optimization module, the network configuration controller determines the suppliers from which to purchase raw material and the customers to sell the product to by receiving real time or daily data regarding the network design parameters from the previously selected plants and warehouses and all the available suppliers and customers. A Linear Programming (LP) model for the daily supply chain network design is then formulated.

This LP model is basically reduced from the two-stage stochastic programming in Equations (6-1)-(6-12) after the first stage decision variables (plant and warehouse decisions) have been fixed. Unlike the decisions pertaining to plant and warehouse locations, this decision can be changed frequently because it is inexpensive to change suppliers and/or customers. To determine the optimal supplier and customer portfolio, the following Linear Programming (LP) model is solved.

Indices and sets:

i, j = index of nodes in the supply chain

S = set of candidate suppliers

P = set of selected plants

W = set of selected warehouses

C = set of customers

$F = P \cup W$

$N_d = S \cup F \cup C$

A = set of arcs between nodes

Model parameters:

q_{ij} = unit shipping cost of product flowing from node i to node j

d_j = demand at node j

s_i = available supply at node i

r_j = per-unit capacity required when a product is processed at plant or warehouse j

m_j = capacity of facility j

Decision variables:

x_{ij} = units of product flowing from node i to node j. X = vector of x_{ij} where i, j ∈ N_d

$$\text{Min} \quad \sum_{(ij)\in A} q_{ij}\, x_{ij} + \sum_{j\in C} g_j u_j \tag{7-1}$$

$$s.t. \quad \sum_{i\in N_d} x_{ij} - \sum_{l\in N_d} x_{jl} = 0 \;; \qquad \forall j \in F \tag{7-2}$$

$$\sum_{i\in W} x_{ij} + u_j \geq d_j \;; \qquad \forall j \in C \tag{7-3}$$

$$\sum_{j\in P} x_{ij} \leq s_i \;; \qquad \forall i \in S \tag{7-4}$$

$$r_j\left(\sum_{i\in N_d} x_{ij}\right) \leq m_j \;; \qquad \forall j \in F \tag{7-5}$$

$$X \in R_+^{|A|}\,,\; U \in R_+^{|C|} \tag{7-6}$$

The objective function in Equation (7-1) is to minimize the sum of the daily shipping cost and penalty cost of the supply chain. Equation (7-2) is the flow conversation constraints on the facility nodes. The demand constraints for customer nodes are given in Equation (7-3). The supply constraints are given in Equation (7-4). Equation (7-5) expresses the capacity constraints at the facility nodes. Lastly, the non-negative variable constraints are stated in Equation (7-6). As we can see, the daily network design model being solved by the network configuration controller is an LP problem that is much easier to solve than the large two-stage stochastic programming solved by the stochastic network optimization module. Therefore, the daily optimization approach in this controller is certainly practical for real life supply chains.

The result obtained from the network configuration controller is the selected suppliers and customers on the current date. By combining this result with the plant and warehouse locations obtained from the stochastic network optimization module, we have all the information necessary for an entirely new configuration of the supply chain network. Though simple conceptually, network reconfiguration presents some significant challenges in practice. First, to ensure the best performance of the reconfigurable supply chain network, it is necessary to find as many supplier and customer nodes as possible (this ability is called discoverability). In traditional supply chains, searches for suppliers or customers are often conducted manually by purchasing or marketing departments respectively. To obtain the most updated and complete list of suppliers and customers, this search should be able to be done over the Internet such that potential suppliers and customers worldwide will be included in the pool of potential supply chain partners. Second, even though we have the most updated and complete list of supply chain partners, making transaction with all possible partners is likely to be unrealistic assuming that the trading partners run their businesses on diverse enterprise software applications. Communicating with those partners requires different communication protocols depending on which enterprise software platforms they are using. Therefore, we still need an approach that allows us to address this issue of interoperability. The approach to this issue is discussed next.

7.2.2 SOA for a reconfigurable supply chain network

Erl (2005) defines SOA as "a form of technology architecture that adheres to the principles of service-orientation. When realized through the Web services technology platform, SOA establishes the potential to support and promote these principles throughout the business process and automation domains of an enterprise." And, this is the definition adopted for the present study. Specifically, each node in the supply chain can be represented as a service. When applied to a supply chain network design, the principle of service-orientation allows network reconfiguration to take place due to the former's properties of discoverability and interoperability. As previously stated in Section 7.2.1, these two properties are required in order to implement a reconfigurable supply chain network. Therefore, our reconfigurable supply chain network is designed based on the SOA principle. The important properties of the service-oriented design, stated in (Erl, 2005), were given in Section 3.6 and are repeated here:

1. *Loose coupling: services maintain a relationship that minimizes dependencies and only requires that they retain an awareness of each other.*
2. *Service contract: Services adhere to a communications agreement, as defined collectively by one or more service descriptions and related documents.*
3. *Autonomy: Services have control over the logic they encapsulate.*
4. *Abstraction: Beyond what is described in the service contract, services hide logic from the outside world.*
5. *Reusability: Logic is divided into services with the intention of promoting reuse.*
6. *Composability: Collections of services can be coordinated and assembled to form composite services.*
7. *Statelessness: Services minimize retaining information specific to an activity.*
8. *Discoverability: Services are designed to be outwardly descriptive so that they can be found and assessed via available discovery mechanisms*

Although not explicitly stated, as a by-product of loose coupling and abstraction, interoperability can be achieved via a service-oriented design. In our work, the supply chain information system that has the eight properties stated above is developed to enable network reconfiguration ability in the presence of risk.

In this work, each node in the supply chain can perform as both a service provider and requester. For example, a plant is considered a service provider to a warehouse. On the other hand, it can be also considered a service requester to a supplier. Therefore, the role that a node performs depends on the business process in which it is involved. In addition, each node is considered *autonomous* since it has its own logic for operating its business processes. Figure 7-2 gives a brief view in schematic terms of the SOA-based reconfigurable supply chain network.

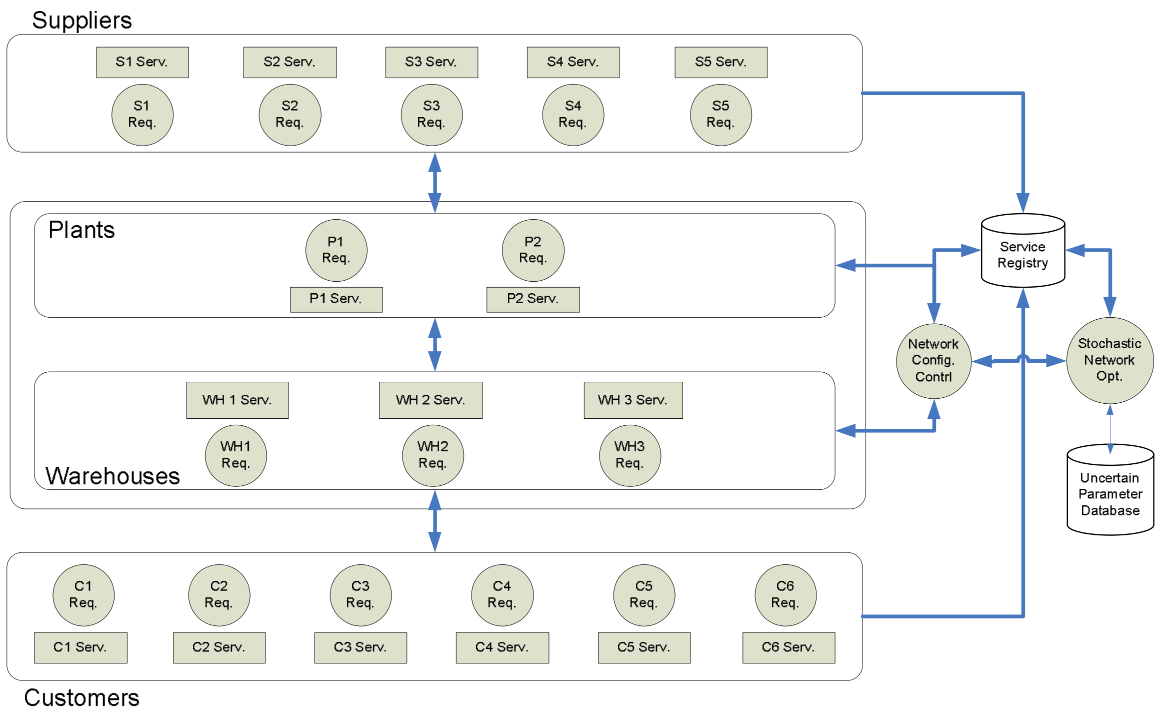

Figure 7-2: Overview of the SOA-based reconfigurable supply chain network

Figure 7-2 shows that each node in the supplier, plant, warehouse, and customer layers comprises a service provider that can be deployed with the *statelessness* property using Web services technology (shown as a rectangle) and a service requester (shown as a circle). Based on the SOA principles, a node in one layer can freely communicate with any node in another layer due to *loose coupling* and *abstraction* properties of the network. Hence, to deliver a product to a customer, the supply chain can compose and reuse the services from any warehouse, plant, and supplier to form a product chain to the customer. This certainly represents the *composability* and *reusability* properties of the supply chain. Another four components are included in this architecture. First, the stochastic network optimization module discussed in Chapter 6 is responsible for determining the locations of plants and warehouses. Second, the uncertain parameter database from Chapter 5 is used as a storage for empirical distributions of the uncertain network design parameters. Third, the network configuration controller presented in Section 7.2.1 is responsible for activating the selected plants and warehouses and determining the daily network configuration. Last of all, the service registry contains the most updated and complete list of the available nodes from which the supply chain can be formed. This service registry basically enables *discoverability* in the supply chain. This supply chain information architecture is designed such that its communication sequence follows the procedure to develop the reconfigurable supply chain network, as shown in Figure 7-1.

The next chapter details the deployment of and technologies used in this reconfigurable network and assesses the system's performance. It also discusses how to obtain the *service contract* property in the proposed architecture.

Chapter 8

Deployment and Assessment of the SOA-Based Reconfigurable Supply Chain

8.1 Deployment of the SOA-based reconfigurable supply chain

The SOA-based reconfigurable supply chain network can be realized by using existing and soon-to-come technologies and standards in practice. The main enabling technology for the reconfigurable supply chain is the Web services technology. In order to fulfill the eight properties of the service-oriented design in Section 7.2.2, only three main standards are required: Web Services Description Language (WSDL) (http://www.w3.org), SOAP (http://www.w3.org), and Universal Description Discovery and Integration (UDDI) (http://www.oasis-open.org). Figure 8-1 illustrates the relationships among these three Web services standards.

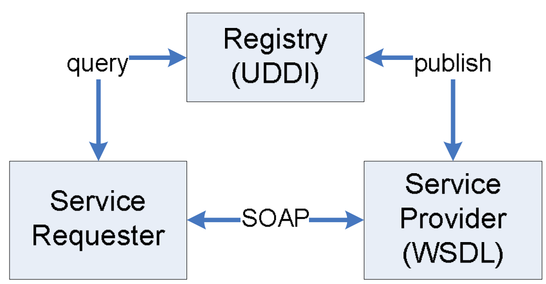

Figure 8-1: Web services standards

According to Figure 8-1, a service provider publishes its service on a service registry using the UDDI standard, which renders its service visible to other service requesters. Next a service requester queries a service from the service registry and the registry returns a list of associated service providers and their addresses (URLs) to the requester. The requester then contacts the selected provider to perform a transaction. In order for the requester to communicate with the provider, the requester must follow the provider's communication protocol. This is where WSDL comes in. WSDL is an XML-based service description that is defined by a service provider. To initiate communication, the requester must first find the targeted provider according to the obtained URL; next, having found the provider, the requester reads the provider's WSDL and creates a message that comply with this WSDL to communicate with the provider. The messages exchanged between the provider and the requester are created in the SOAP format. Since both WSDL and SOAP are XML-based documents, the communication in the supply chain can be done seamlessly and interoperability can be achieved.

Figure 8-2 is a more detailed version of Figure 7-2 presented earlier. It shows how the SOA-based reconfigurable supply chain network can be deployed in practice. In the rest of this section, we will discuss the realization of each component in detail following the numbers labeled in Figure 8-2.

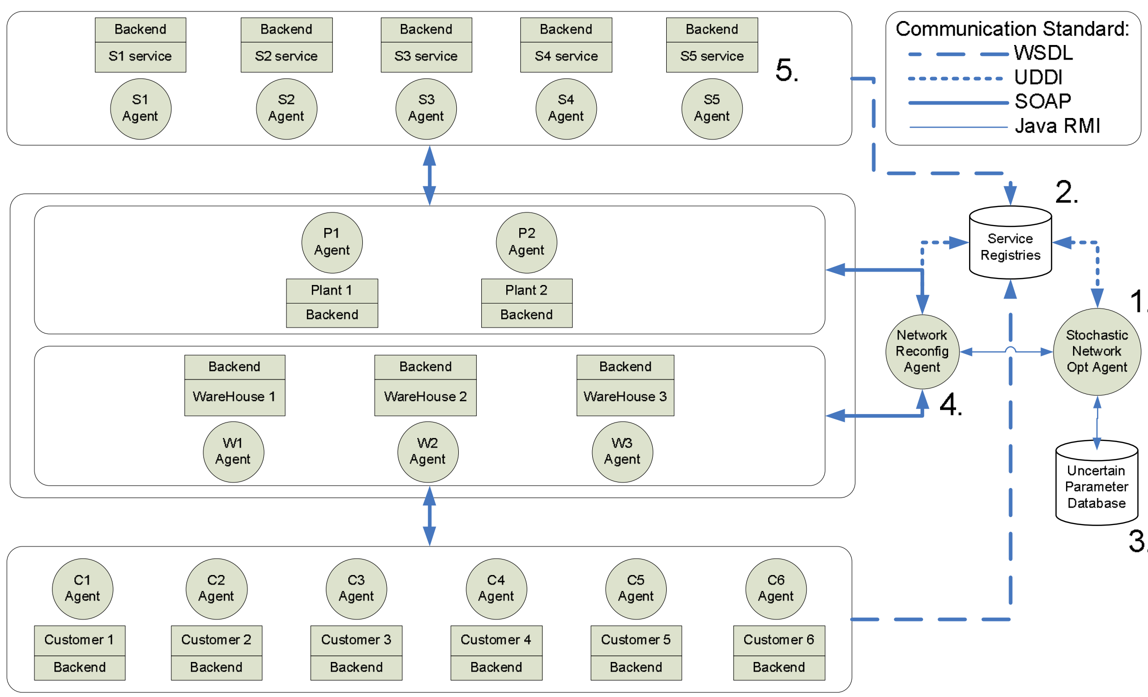

Figure 8-2: Overview of the deployment of the SOA-based reconfigurable supply chain network

8.1.1 Stochastic network optimization agent

This optimization unit is used to determine the long-term configuration in the plant and warehouse layers of the supply chain as explained in Chapter 6. To implement this unit, we formulate the two-stage stochastic programming model in Equations (6-1)-(6-12) in Java with the ILOG Concert Technology for Java API (http://www-01.ibm.com) and receive a list of available nodes from the service registry and probability distributions of the network design parameters from the uncertain parameter database. Then we solve the problem using CPLEX 11.0. After obtaining a solution of plant and warehouse locations, this solution is sent to the network configuration controller.

8.1.2 Service registry

The service registry contains a list of services and their addresses available in the dynamic environment. UDDI is commonly regarded as the most appropriate standard for implementing the service registry. The UDDI specification has been successfully defined by OASIS; however, its implementation in practice has never been announced formally in both business and academia. The best we have seen so far is efforts that center on running prototype versions of public UDDI registries by some of the large IT companies, including Microsoft, IBM, and SAP. Apparently, none of them is actively supported now. One of the most challenging aspects of building a public UDDI registry is concerned with semantics. Since different services may use different ontologies when publishing information and searching for other services, the results obtained from a UDDI registry may not be fully correct. Therefore, semantics and ontology are considered one of the most important research areas in the future for UDDI realization.

Although public UDDI registries are not readily available, private UDDI registries are deployed in some organizations for internal use. In this research, we decide to create our own simple service registry to hold services in our reconfigurable supply chain network. This service registry can perform the function of a yellow pages service in a way that though similar to the UDDI registry is less complex. Basically, we create this service registry as a web service itself using an Enterprise Java Bean (EJB) component available in Java EE platform (http://java.sun.com). This EJB is subsequently connected to a database that holds a list of all the available supply chain services and their addresses. By creating the service registry as a web service, the other nodes in the supply chain can easily connect to the registry via the SOAP messaging standard.

8.1.3 Uncertain parameter database

The uncertain parameter database contains the empirical probability distributions of the uncertain network design parameters. In order to obtain a set of data points for creating the empirical distributions, the Multi-Agent-Based Simulation was developed in JAVA with the Java Agent DEvelopment (JADE) API (http://jade.tilab.com/) and the simulation experiments were carried out to observe the shipping costs, supplies, demands, and facility capacities as discussed in Chapter 5. These empirical distributions are then used as inputs to the two-stage stochastic programming model in Equations (6-1)-(6-12).

8.1.4 Network configuration controller

The network configuration controller discussed in Section 7.2.1is implemented in Java. The LP model in Equations (7-1)-(7-6) is formulated and coded using LINDO API 6.0 for Java (http://www.lindo.com/). The result obtained from this LP model is a list of selected suppliers and customers, as well as quantities shipped between nodes in the supply chain. These data will be sent to the previously selected plant and warehouse nodes to form the daily supply chain network accordingly.

8.1.5 A node in the supply chain

The component labeled no. 5 represents a node in the supply chain. The structures of all the nodes in the supply chain are the same and consist of three main subcomponents:

- Service requester: every node in the supply chain contains a service requester that asks for required services from the other nodes. For example, if a plant needs to replenish its raw material inventory, a service requester will query a list of suppliers from the service registry and then contact raw material supply services at supplier nodes. In the Java EE platform, a service requester can be deployed as a Java application client or Java Web client (Java Servlet or JavaServer Pages (JSP)). In our architecture, we developed service requesters as the Java application clients.
- Service provider: each node in the supply chain also contains a service provider subcomponent to perform its service for the other nodes. A service provider is commonly deployed as a Web service in the supply chain. For example, each plant in Figure 8-2 can perform a manufacturing service, such that any warehouse can purchase products from it. To complete the transaction, a warehouse needs to contact a Web service at a plant; as a result, the plant is considered as a service provider in this case. In addition, a service provider is also able to perform some local reasoning based on its own logics, e.g. how many units of raw material/product it needs to procure on a specific date or the maximum unit price that can be accepted when purchasing the raw material/product. The stochastic

network optimization agent and the network configuration controller in Section 8.1.1 and 8.1.4 only suggest the network configuration at a strategic level, i.e. facility locations and supplier and customer selections. However, the service provider still needs to determine some local decisions. For example, it still needs to determine the daily quantity of raw material/product to procure using its inventory forecasting technique (in this work, the nodes use the order up-to-level policy). Also, it may need to execute some sophisticated reasoning algorithms in order to determine the maximum unit price accepted when purchasing the raw material/product (however, to simplify the simulation model in this work, it is assumed that the maximum price is infinite). Other local reasoning includes daily production planning, real time product pricing, daily order allocation to customers, etc. By retrieving associated input data from the backend system, the service provider can autonomously and independently perform the aforementioned local reasoning based on its own logics. To deploy a Web service in the Java EE platform, either Java Web components (Servlet, JavaServer Pages (JSP), JavaServer Faces, Java RESTful web services) or Enterprise Java Bean (EJB) can be used. In our work, we deploy a service provider using the EJB because of its ability to connect to an enterprise's database or backend system.

- Backend system: a backend system is built in every node in the supply chain. It is responsible for storing data of an enterprise. A backend system allows a separation between the service logic and enterprise data. A service does not need to keep records of its processed data, and as a result, the property of *statelessness*, as referenced in Section 7.2, can be achieved. In practice, a backend system can be just a single-database application or it could be a more sophisticated Enterprise Information System (EIS) such as a Material Requirements Planning (MRP) or Enterprise Resource Planning (ERP) system. In our work, we implement the backend system at each supply chain node as a simple MySQL 5.1 database server (http://www.mysql.com/).

In the next section, we develop a simulation model to demonstrate mechanics of the reconfigurable supply chain and assess the benefits that accrue from its reconfigurability.

8.2 Performance assessment using simulation experiments

In order to assess the benefits obtained by applying a network reconfiguration strategy in a risk environment, we developed an agent/SOA-based simulation model to observe potential savings and the extent to which the reconfigurable supply chain mitigates risk. The architecture of this reconfigurable supply chain is similar to that shown in Figure 8-2. Two modeling approaches were combined and used in this simulation model: software agent and web services. It should be noted here that this agent/SOA-based reconfigurable supply chain simulation model is not to be confused with the Multi-Agent-Based Simulation (MAS) model discussed in Chapter 5. Such a MAS model is used to obtain the empirical distributions of the uncertain network design parameters and then store those distributions in the uncertain parameter database, whereas the agent/SOA-based simulation model is used to assess the performance of the proposed architecture for the reconfigurable supply chain network.

The software agent approach was used to model the stochastic network optimization module, the network configuration controller, and the service requester in every supply chain node. However, the Web services approach was used to deploy the service registry and the service provider at every node in the supply chain. Particularly, the agents were created using JADE Java API, and the Web services were built using Java API for XML Web Services (JAX-WS) in the Java EE platform. To deploy the backend systems, we built a MySQL database instance for each supply chain node. Lastly, a Java object was created to represent the uncertain parameter database and is responsible for fitting the empirical distributions (from the data obtained from the MAS model in Chapter 5) and sends these distributions to the stochastic network design module.

The messaging between the uncertain parameters database, the stochastic network optimization module, and the network configuration controller are carried out using Java RMI. Service providers publish their services on the service registry in WSDL format, and the service requesters and providers communicate via SOAP messaging. The only different messaging format between this agent/SOA-based simulation model and the architecture proposed in Section 8.1 (shown in Figure 8-2) inheres in the messaging from the stochastic network optimization module and the network configuration controller to the service registry. Because there is no official public UDDI registry to date, we deploy the service registry in our model as a web service. So, all inquiries and publishing to this registry take place through SOAP messages only.

8.2.1 Performance measures

To assess the performance of the reconfigurable supply chain network, two performance measures were used in the simulation experiments:

Total cost reduction (%): this measure indicates how much can be saved by using the network reconfiguration strategy compared to not using the strategy in the presence of risk. The mathematical expression is shown in Equation (8-1):

$$Cost\ reduction = (C_s - C_r) \times 100/C_s \tag{8-1}$$

where C_s = total cost consisting of facility, product shipping, inventory holding, and product shortage costs of a static supply chain (without reconfiguration ability) under risk condition, and C_r = total cost including facility, product shipping, inventory holding, and product shortage costs of a reconfigurable supply chain under risk condition.

Risk mitigation index: this measure indicates the extent to which the network reconfiguration strategy can mitigate the risk. In particular, this measure compares the total costs of the static supply chain with no-risk and the reconfigurable supply chain under risk condition. The measure can be expressed as in Equation (8-2):

$$Risk\ mitigation\ index = C_{nr}/C_r \tag{8-2}$$

where C_{nr} = total cost including facility, product shipping, inventory holding, and product shortage costs of a static supply chain (without reconfiguration ability) under no-risk condition, and C_r is similar to Equation (8-1). Ideally, we would want the risk mitigation index to be greater than or equal to one. If this is the case, it means that the reconfigurable supply chain can completely mitigate the risk in terms of the total supply chain cost. In other words, if the risk mitigation index is equal to one, it means that the reconfiguration strategy can mitigate the risk in the supply chain such that its financial performance is maintained at the same level as the one under no-risk condition. Furthermore, if the risk mitigation index is greater than one, this indicates that the reconfigurable supply chain under risk condition can even result in better performance than the static supply chain under no-risk condition from financial perspective. On the other hand, if the index is much lower than one, e.g. less than 0.5, we can interpret that the reconfiguration strategy fails to mitigate the risk in this supply chain.

8.2.2 Problem settings

To assess the performance of the reconfigurable supply chain network, simulation experiments were run on three example problems. In these experiments, we considered the supply chain network and its associated data shown in Figure 2-1 and Table 2-1 as our base case (the complete problem data can be found in Appendix F). Then we applied three types of risk scenarios to this supply chain, i.e. no-risk, high-long-supplier risk, and high-long-plant risk. Then we computed the total cost reduction measure and risk mitigation index for the two risk scenarios when the network reconfiguration strategy is applied. The experiments were run on the developed agent/SOA based simulation model with ten replications for each. The problem settings and experimental results are given below.

8.2.2.1 No-risk

The first case is a problem in which there is no risk in the supply chain. The problem scenario in this case is similar to the problem in Figure 2-1 and Table 2-1. From the data given in Figure 2-1 and Table 2-1, we solved the MIP model of Equations (2-1) –(2-7) to determine the supply chain configuration in a no-risk scenario. The resulting supply chain is expressed in Figure 2-2. Then we built and ran a simulation experiment based on this network configuration and recorded its total supply chain cost.

8.2.2.2 High-frequency-long-duration-supplier risk

In the second case, we inserted a high-frequency and long-duration risk event at supplier 3 in Figure 2-1. This risk event is assumed to randomly occur six times in one year. Also, once it has occurred, the risk duration will be uniformly distributed from between 13 and 15 days; in addition, the risk event occurs at supplier 3 only. To run the experiment, we built the agent-based simulation models for two types of supply chains:

Static supply chain with no reconfiguration ability: SOA is not applied. The network configuration is based on Equations (2-1)-(2-7) and the nodes selected are:

- Suppliers 3 and 5
- Plant 7
- Warehouses 9 and 10

- Customers 11, 12, 13, 14, 15, and 16

Reconfigurable supply chain with reconfiguration ability: SOA is applied. The network configuration is based on Equations (6-1)-(6-12) and the nodes selected are:

- Supplier: varied
- Plant 7
- Warehouses 9 and 10
- Customers 11, 12, 13, 14, 15, and 16

Then we recorded the total supply chain costs for both model types.

8.2.2.3 High-frequency-long-duration-plant risk

In the last case, the high-frequency and long-duration risk event was inserted into plant 7 in Figure 2-1. This risk event occurs at plant 7 six times a year and, once it has occurred, it lasts for 13–15 days (uniform distribution). Then, we again built the agent-based simulation models similar to those discussed in Section 8.2.2.2.

Static supply chain with no reconfiguration ability: SOA is not applied. The network configuration is based on Equations (2-1)-(2-7) and the nodes selected are:

- Suppliers 3 and 5
- Plant 7
- Warehouses 9 and 10
- Customers 11, 12, 13, 14, 15, and 16

Reconfigurable supply chain with reconfiguration ability: SOA is applied. The network configuration is based on Equations (6-1)-(6-12) and the nodes selected are:

- Supplier: varied
- Plant 6
- Warehouses 9 and 10
- Customers 11, 12, 13, 14, 15, and 16

8.2.3 Experimental results and discussion

Table 8-1: Experimental results

			High-long-supplier risk event		High-long-plant risk event	
Cost	**Node**	**Static-no risk**	**Static-risk**	**Reconfig.-risk**	**Static-risk**	**Reconfig.-risk**
Facility cost	Plants	400,000.0	400,000.0	400,000.0	400,000.0	500,000.0
	Warehouses	195,000.0	195,000.0	195,000.0	195,000.0	195,000.0
Base operating cost	Plants	720,000.0	720,000.0	720,000.0	720,000.0	900,000.0
	Warehouses	360,000.0	360,000.0	360,000.0	360,000.0	360,000.0
Shipping cost	Suppliers	151,895.7	113,344.4	170,293.8	115,629.7	141,915.0
	Plants	429,515.4	319,640.3	436,256.8	340,396.0	531,773.0
	Warehouses	965,017.3	695,856.2	1,010,800.8	741,731.8	1,117,686.1
Inventory cost	Plants	44,683.6	32,638.5	51,808.4	51,668.0	44,538.4
	Warehouses	14,197.0	12,028.1	13,276.3	12,536.3	5,283.9
Shortage cost	Customers	965,970.0	3,433,630.0	218,300.0	2,995,650.0	0.0
Total		4,246,279.0	6,282,137.5	3,575,736.1	5,932,611.8	3,796,196.4

From Table 8-1, five experiments were carried out, i.e., one in a no-risk scenario, two from a high-frequency-long-duration-supplier risk event, and another two from a high-frequency-long-duration-plant risk event. As we can see, in the no-risk scenario, only the static supply chain was observed for its total cost. However, for the risk scenarios, both static and reconfigurable supply chains were tested for their performance. In each experiment, we observed five cost components including facility, base operating, shipping, inventory, and shortage costs, as shown in Table 8-1. From Table 8-1, we can see that the results are as expected. First, the total costs in the static supply chain under risk scenarios are higher than the total cost in the no-risk scenario. Next, we see that the total cost of the static supply chain in the high-long-supplier risk event and the high-long-plant risk event scenarios are very close. This is expected because the supplier risk is more serious in the sense that the risk occurred at the upstream of the supply chain. However, in terms of production throughput, plant 7 has a high throughput and it is the only manufacturing site in the network. Therefore, the trade-off between node location and production throughput makes the total costs from these two experiments very close. A further point of importance is that the total costs of the reconfigurable supply chain in the two risk scenarios are lower than the total cost of the static supply chain in the no-risk case. The reason for this improvement is discussed next. Based on Equations (8-1) and (8-2), the performance measures are computed and shown in Table 8-2.

Table 8-2: Performance measures

High-long-supplier risk event	Cost reduction	43.08%
	Risk mitigation index	1.19
High-long-plant risk event	Cost reduction	36.01%
	Risk mitigation index	1.12

According to Table 8-2, the cost savings that accrued from applying the network reconfiguration strategy are 43.08% and 36.01% for the high-long-supplier risk event and the high-long-plant risk events, respectively. For the risk mitigation index, the reconfigurable supply chain network yielded indices greater than one in both risk scenarios: 1.19 and 1.12 for the high-long-supplier risk event and high-long-plant risk event, respectively. The reason for this is that, in the reconfigurable network, the supply chain is able to adaptively procure from and sell to the most appropriate supplier and customer portfolio. This adaptability basically makes the supply chain more agile under risk conditions. In other words, agility can make small, but still desirable, improvement in the total cost of this supply chain even though no risk event has occurred or is imminent.

Chapter 9

Conclusions and Future Work

Supply chain risk management is indispensable in today's stochastic market places. At the strategic level, firms must consider stochastic behavior of the environment when designing their supply chain networks. In this research, we proposed a two-stage stochastic programming model that can be used to determine a supply chain network configuration in the presence of risk. In addition, with the aid of advanced information technologies, this stochastic programming model is integrated into the proposed information system architecture allowing the supply chain to be reconfigurable and more tolerant to the potential risk.

In particular, this research proposed an architecture for a reconfigurable supply chain network subject to risk. The main advantage of this architecture is its ability to reconfigure itself once a risk event to the supply chain has been predicted. The new network configuration is more capable of withstanding a supply chain risk event; that is, the impact of a risk event can be planned for and mitigated. The reconfigurable supply chain network consists of three main parts, as shown in Figure 4-1:

1. The uncertain parameter database: this component stores knowledge about the effects of risk events on a supply chain network. In particular, this knowledge is stored as probability distributions of the network design parameters, including shipping costs between nodes, demand at each customer node, amount of supply at each supplier node, and facility capacities, for each type of risk. In this research, supply chain risks are categorized into 36 types based on their frequency, duration, and location. Therefore once a risk event has occurred, it is matched to one of these 36 risk types and the probability distributions can be retrieved appropriately.

2. The stochastic network optimization: this component is responsible for determining configuration at the plant and warehouse layers. To determine the plant and warehouse locations, a redesign problem of a supply chain network under risk is formulated as a two-stage stochastic programming model and solved using the Sample Average Approximation (SAA) method, L-Shaped decomposition, and efficient sampling techniques. To generate the random parameters in the two-stage stochastic programming model, the risk type realized in the supply chain is matched to one of the 36 types in the uncertain parameter database. Then the associated probability distributions are used to generate the random parameter values accordingly. Lastly, the stochastic network optimization sends the decision on plant and warehouse locations to the network configuration controller.
3. The network configuration controller: this component receives the location data of plants and warehouses from the stochastic network optimization, and then it activates the operations of such plants and warehouses. In addition, it is also responsible for determining suppliers to procure from and customers to serve on a daily basis. In order to determine this daily supply chain network configuration, the problem is formulated in an LP model and solved in order to obtain a list of suppliers and customers on a daily basis.

Two central challenges exist in deploying this reconfigurable supply chain network: the achievement of discoverability of the supply chain nodes and of interoperability among all nodes in the supply chain. To address these challenges, Service-Oriented Architecture (SOA) was applied to designing the network architecture. The service registry in the proposed architecture facilitates the discoverability in the SOA-based reconfigurable supply chain. Whereas, SOAP messaging, an XML-based message system, enables interoperability among nodes in the supply chain; however, by drawing on the principle of service orientation, the SOA-based supply chain is able to adaptively change its configuration to be more robust against a risk event that is occurring in its immediate environment.

Through the use of current Web services technology and standards, including WSDL, SOAP, and UDDI, the deployment of the SOA-based reconfigurable is possible, and the benefits obtained from adopting this architecture design were demonstrated by the simulation experiments. The first performance measure recorded in the simulation experiments is the Total Cost Reduction, which indicates the total annual savings of the entire supply chain cost when the SOA-based reconfigurable supply chain is applied in a risk environment. The second measure is the Risk Mitigation Index, which indicates the extent to which the network reconfiguration strategy can mitigate the risk. In other words, this measure compares the annual total supply chain cost of a no-risk environment with that of a risk environment. Ideally, we would want the annual total cost of the supply chain when a risk event has occurred to be equal to or very close to the total cost when no risk event has occurred. According to the results of the simulation experiments, both performance measures significantly improve when the SOA design is applied to the supply chain network.

Although the results from the simulation experiments proved satisfactory, deploying the SOA-based reconfigurable supply chain in practice still requires further research as follows:

- First, at the time of writing, there has been no formal or reliable public UDDI registry available to industry. Some efforts have been made by large IT companies, including Microsoft, IBM, and SAP, to set up prototypes of the public UDDI registries; however, they have been discontinued and their futures are still unknown. Due to the lack of a UDDI registry, several organizations currently implementing SOA have developed their own private UDDI registries to deploy Web services for internal use. This solution approach promotes a service-oriented environment within an organization, but not for inter-organization purposes as required for developing a reconfigurable supply chain network. Therefore, to realize a reconfigurable supply chain network, a reliable public UDDI registry should be established.

- Second, even if a public UDDI registry were set up, there is still an issue regarding semantics and ontology that makes publishing information about services and querying services in a UDDI registry problematic. To enable service discovery in a supply chain, participating companies must agree on the ontology used at least within their industry.
- Third, performing transactions online can be fast and precise, but for Web services and SOAP messaging, security issues are still of concern to users. The proposed architecture in this study did not consider the security aspects of an SOA-based supply chain. In order to deploy this architecture in practice, a security policy would have to be developed and included in its design. The current WS-Security specification, published by OASIS and supported by Java's Metro Web Services project from Java, could be a good candidate for enabling SOAP message security in an SOA-based reconfigurable supply chain. Specifically, how to integrate this specification into a reconfigurable supply chain is one of the challenging topics in the future research.
- Finally, it is assumed in this work that all business transactions with suppliers and customers are conducted online without human intervention. In practice, this assumption may be unrealistic for high-value transactions. It is a fact that manual transactions with more intensive approval levels from management is still preferred when negotiating high-value contracts or transactions. Therefore, it is desirable that dedicated online negotiation protocol should be developed to support such high-value transactions in the future.

As a last note, the development of the proposed reconfigurable supply chain network is based on several assumptions in the problem description in Chapter 2, the Multi Agent-based

Simulation (MAS) model in Chapter 5 and the two-stage stochastic programming model in Chapter 6. Relaxing or changing some of these assumptions may result in invalidity of the proposed framework. In this research, we divide these assumptions into two main categories:

- Problem settings: the problem settings are the assumptions that we can change without affecting the validity of the proposed reconfigurable supply chain network framework. These settings mainly include operational parameter values of the MAS model. All the assumptions in Appendix A except 1, 7, and 8 fall in this category. Similarly, some assumptions for the two-stage stochastic programming model in Section 6.1 can be considered in this category including assumption 6 and 7. These problem settings are basically input parameters to the reconfigurable supply chain network. If it is required to change any of these assumptions, we can easily change its value while maintaining the validity of the framework.
- Problem restrictions: the problem restrictions are the assumptions whose changes may result in invalidity of the reconfigurable supply chain network framework. These restrictions include 1) the commodity and low value product assumptions in Section 2.1, 2) Assumption 1, 7, and 8 in Appendix A, 3) Assumptions 1 to 5 in Section 6.1, and 4) uncertainty only in unit shipping cost, customer demand, supply availability, and facility capacity. For restriction 1, if the commodity product restriction is relaxed, the supply chain may not be able to switch from an existing supplier to a new one easily because different suppliers provide different product characteristics and the similar reason also holds for the customer side. Therefore, in this case, we need to extend our framework such that the product characteristics can be measured in terms of its functions and quality and these characteristics can be used as the criteria to select new suppliers. In addition, if the low value product restriction is relaxed, a more secured transaction protocol should be developed to support the transaction of high value products in the SOA architecture. For

restriction 2, if assumption 1 or 7 is changed, we may not be able to obtain the probability distributions of the network design parameters correctly, consequently, the solution to the two-stage stochastic programming may not be accurate. In addition, if assumption 8 is changed, the mathematical expression of the two-stage stochastic programming model must be changed as well. Similarly, if assumption 1 to 5 of restriction 3 are altered, the structure of the two-stage stochastic programming model needs to be modified according to the new restriction. Lastly, if restriction 4 is relaxed, i.e. some network design parameters other than shipping cost, customer demand, supply quantity, and facility capacity are random, the two-stage stochastic programming problem may not be able to solve using the L-shaped method anymore since only the right hand side of Equation (6-7)-(6-11) are allowed to be random when solving the problem using this method.

In summary, while the proposed reconfigurable supply chain network framework are flexible such that ones can alter its problem settings, it must be ensure that the problem considered still conforms to the restrictions discussed above.

References

- Agrawal, V. and S. Seshadri. (2000). Risk intermediation in supply chains. IIE Transactions, 32, 819-831.
- Antonov, I.A. and V.M. Saleev. (1980). An economic method of computing LPτ – sequence. USSR Computational Mathematics and Mathematical Physics, 19, 252-256.
- Bellifemine, F., G. Caire, A. Poggi and G. Rimassa. (2003). Jade a white paper, exp. 3(3), 6-19.
- Bertsimas, D. and A. Thiele. (2006). A robust optimization approach to inventory theory. Operations Research, 54(1), 150-168.
- Billington, C. (2002). HP cuts risk with portfolio approach. Purchasing, February 21, 2002, 43-45.
- Birge, J.R. and F. Louveaux. (2000). Introduction to stochastic programming. Springer.
- Bogataj, D. and M. Bogataj. (2007). Measuring the supply chain risk and vulnerability in frequency space. International Journal of Production Economics, 108, 291-301.
- Chan, F.T.S. and H.K. Chan. (2005). Simulation modeling for comparative evaluation of supply chain management strategies. International Journal of Advanced Manufacturing Technology, 25, 998-1006.
- Chopra, S. and M.S., Sodhi. (2004). Managing risk to avoid supply-chain breakdown. MIT Sloan Management Review, 46 (1), 52-61.
- Chopra, S., G. Reinhardt and U. Mohan. (2007). The importance of decoupling recurrent and disruptions risks in a supply chain. Naval Research Logistics, 54 (5), 544-555.
- Deleris, L.A. and F. Erhun. (2005). Risk management in supply networks using Monte-Carlo simulation. Proceeding of the 2005 Winter Simulation Conference, 1643-1649.

- Desroches, C.M., R.J. Blendon, and J.M. Benson. (2005). Health Affairs. Chevy Chase, 24 (3), 822-831.
- Dietrich A.J., S. Kirn and V. Sugumaran. (2007). A service-oriented architecture for mass customization-a shoe industry case study. IEEE Transactions on Engineering Management, 54(1), 190-204.
- Eppen, G.D., R.K. Martin and L. Schrage. (1989). A Scenario approach to capacity planning. Operations Research, 37(4), 527-527.
- Erl, T. (2005). Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall PTR.
- Foroughi, A., M. Albin, and M. Kocakulah. (2006). Perspectives on global supply chain supply-side risk management. PICMET 2006 Technology Management for the Global Future, 2732-2740.
- Gaonkar, R. and N. Viswanadham. (2004). A conceptual and analytical framework for the management of risk in supply chains. Proceeding of the 2004 IEEE International Conference on Robotics & Automation, 2699-2704.
- Gaonkar, R.S. and N. Viswanadham. (2007). Analytical framework for the management of risk in supply chains. IEEE Transactions on Automation Science and Engineering, 4(2), 265-273.
- Gentle, J.E. (2004). Random number generation and Monte Carlo methods. Springer.
- Günüs, A.T. and A.F. Güneri. (2007). Multi-echelon inventory management in supply chains with uncertain demand and lead times: literature review from an operational research perspective. Proceeding of IMechE Part B: Journal of Engineering Manufacture, 221(10), 1553-1570.
- Hallikas, J., I. Karvonen, U. Pulkkinen, V.M. Virolainen and M. Tuominen. (2004). Risk management processes in supplier networks. International Journal of Production Economics,

90, 47-58.

- Han, M. and J. Chen. (2007). Managing operational risk in supply chain. IEEE International Conference on Wireless Communications, Networking and Mobile Computing, 4914-4917.
- Jung, J.Y., G. Blau, J.F. Penkny, G.V. Reklaitis and D. Eversdyk. (2004). A simulation based optimization approach to supply chain management under demand uncertainty. Computers and Chemical Engineering, 28, 2087-2106.
- Kart, F., Z. Shen and C.E. Gerede. (2006). The MIDAS system: a service oriented architecture for automated supply chain management. IEEE International Conference on Service Oriented Computing, 487-494.
- Kleindorfer, P.R. and G.H. Saad. (2005). Managing disruption risks in supply chains. Production and Operations Management, 14(1), 53-68.
- Kleywegt, A.J., A. Shapiro and T. Homem-De-Mello. (2001). The sample average approximation method for stochastic discrete optimization. SIAM Journal on Optimization, 12(2), 479-502.
- Kumar, S., V. Dakshinamoorthy and M.S. Krishnan. (2007). Does SOA improve the supply chain? An empirical analysis of the impact of SOA adoption on electronic supply chain performance. Proceeding of the 40th Hawaii International Conference on System Sciences, 171b.
- Lemieux, C. (2009). Monte Carlo and Quasi-Monte Carlo sampling. Springer.
- Linderoth, J., A. Shapiro and S. Wright. (2006). The empirical behavior of sampling methods for stochastic programming. Annals of Operations Research, 142, 215-241.
- Lucas, C., S.A. MirHassani, G. Mitra and C.A. Poojari. (2001). An application of Lagrangian relaxation to a capacity planning problem under uncertainty. The Journal of the Operations Research Society, 52(11), 1256-1266.

- Mckay, M.D., R.J. Beckman and W.J. Conover. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics, 21(2), 239–245.
- Mele, F.D., G. Guillén, A. Espuña and L. Puigjaner. (2007). An agent-based approach for supply chain retrofitting under uncertainty. Computer and Chemical Engineering, 31, 722-735.
- Nagurney, A., J. Cruz, J. Dong and D. Zhang. (2005). Supply chain networks, electronic commerce, and supply side and demand side risk. European Journal of Operational Research, 164, 120-142.
- Norrman, A. and U. Jansson. (2004). Ericsson's proactive supply chain risk management approach after a serious sub-supplier accident. International Journal of Physical Distribution & Logistics Management, 34 (5), 434-456.
- Özekici, S. and M. Parlar. (1999). Inventory models with unreliable suppliers in a random environment. Annal of Operations Research, 91, 123-136.
- Pai, R.R., V.R. Kallepalli, R.J. Caudill and M. Zhou. (2003). Methods toward supply chain risk analysis. IEEE International Conference on Systems, Man and Cybernetics, 5, 4560-4565.
- Qiu, R.G., Z. Fang, H. Shen, M. Yu and J. Dong. (2007). Design and development of a service-oriented supply chain: an IT perspective. The 5th IEEE International Conference on Industrial Informatics, 585-590.
- Santoso, T., S. Ahmed, M. Goetschalckx and A. Shapiro. (2005). A stochastic programming approach for supply chain network design under uncertainty. European Journal of Operations Research, 167, 96-115.
- Sheffi, Y. (2005). The Resilient Enterprise: Overcoming Vulnerability for Competitive Advantage. Cambridge, MA, The MIT Press.
- Shen, W., Q. Hao, S. Wang, Y. Li, and H. Ghenmiwa. (2007). An agent-based service-oriented

integration architecture for collaborative intelligent manufacturing. Robotics and Computer-Integrated Manufacturing, 23, 315-325.

- Smytka, D. L. and M.W. Clemens. (1993). Total cost supplier selection model: a case study. International Journal of Purchasing and Materials Management, 29, 42-49.
- Snyder, L.V. (2006). Facility location under uncertainty: a review. IIE Transactions, 38, 537-554.
- Snyder, L.V., P.M. Scaparra, M.S. Daskin and R.L. Church. (2005). Planning for disruptions in supply chain networks. Tutorials in Operations Research, 1-23.
- Sobol' I.M. (1967). On the distribution of points in a cube and the approximate evaluation of integrals. USSR Computational Mathematics and Mathematical Physics, 7, 86-112.
- Sodhi, M.S. (2005). Managing demand risk in tactical supply chain planning for a global consumer electronics company. Production and Operations Management, 14 (1), 69-79.
- Stephan, J. and Y. Badr. (2007). A quantitative and qualitative approach to manage risks in the Supply Chain Operational Reference. IEEE the 2nd International Conference on Digital Information Management, 410-417.
- Tang, C.S. (2006). Review: Perspective in supply chain risks management. International Journal of Production Economics, 103, 451-488.
- Teuteberg, F. (2007). Supply chain risk management: a neural network approach. Strategies and Tactics in Supply Chain Event Management 1st edition. Springer.
- Tomlin, B. (2006). On the value of mitigation and contingency strategies for managing supply chain disruption risks. Management Science, 52(5), 639-657.
- Van Slyke, R.M. and R. Wets. (1969). L-Shaped linear programs with applications to optimal control and stochastic programming. SIAM Journal on Applied Mathematics, 17(4), 638-663.
- Wang, D. and Z. Yang. (2007). Risk management of global supply chain. Proceeding of the

IEEE International Conference on Automation and Logistics, 1150-1155.

- Wooldridge, M., N.R. Jennings and D. Kinny. (2000). The Gaia methodology for agent-oriented analysis and design. Autonomous Agents and Multi-Agent Systems, 3, 285-312.
- Zhang, T., S. Ying, S. Cao and X. Jia. (2006). A modeling framework for service-oriented architecture. Proceeding of the Sixth International Conference on Quality Software, 219-266.
- http://jade.tilab.com/. Retrieved on April 15, 2010.
- http://java.sun.com/javaee/. Retrieved on April 15, 2010.
- http://www.fipa.org/. Retrieved on April 24, 2010.
- http://www.iro.umontreal.ca/~simardr/ssj/indexe.html. Retrieved on April 24, 2010.
- http://www.lindo.com/index.php?option=com_content&view=article&id=1&Itemid=9. Retrieved on April 15, 2010.
- http://www.mysql.com/downloads/mysql/. Retrieved on April 15, 2010.
- http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm. Retrieved on April 15, 2010.
- http://www.uml.org/. Retrieved on April 23, 2010.
- http://www.w3.org/. Retrieved on April 22, 2010.
- http://www.w3.org/TR/soap12-part1/#intro. Retrieved on April 15, 2010.
- http://www.w3.org/TR/wsdl20/. Retrieved on April 15, 2010.
- http://www-01.ibm.com/software/integration/optimization/cplex/. Retrieved on April 15, 2010.

Appendix A

Assumptions in A Multi-Agent Based Simulation Model

1. There is only one risk taking place in the network at a time
2. If risk occurs, the production of an associated node will be reduced to zero
3. Due date is always one day after an order is placed
4. Delivery time between each pair of nodes is one day. The products shipped today from node i today will arrive a destination node tomorrow and can be used for production immediately.
5. There is no partial delivery. If a node does not have sufficient quantity of product for an entire order, it will reject an associated RFQ.
6. The unit price of raw material or product on day i is calculated by

$$\begin{aligned} &Unit\ price \\ &\quad = Mean\ price \\ &\quad \times \max(0.7, 0.8 \times \frac{total\ demand\ on\ day\ (i-1)}{Beginning\ inventory\ level\ on\ day\ i - Safety\ stock\ level} \\ &\quad + 0.2 \times \frac{Global\ demand\ day(i-1)}{Global\ supply\ day\ (i-1)} \end{aligned}$$

7. No backordered for unfilled demands because backordered will make demand not independent and identically distributed. This assumption allows us to observe demand distribution correctly
8. There is no direct transportation channel from plant nodes to customer nodes
9. The inventory holding cost per day is 20% of the shipping cost.
10. All nodes forecast demands using the moving average of a 7-day period.
11. Penalty cost for product shortage = 100 unit cost

12. The total operating cost = shipping + holding + penalty cost

Appendix B

Sequence Diagrams for Supply Chain Processes

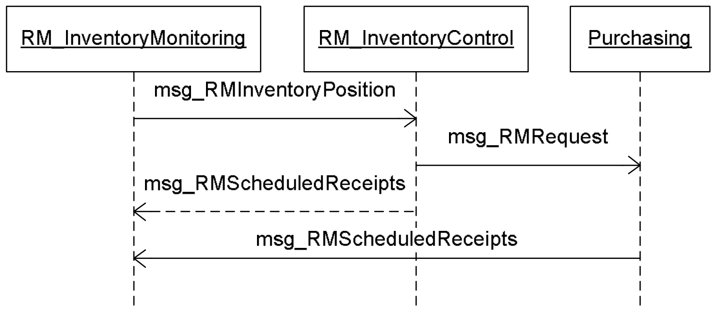

Figure B-1: Sequence diagram for raw material inventory management

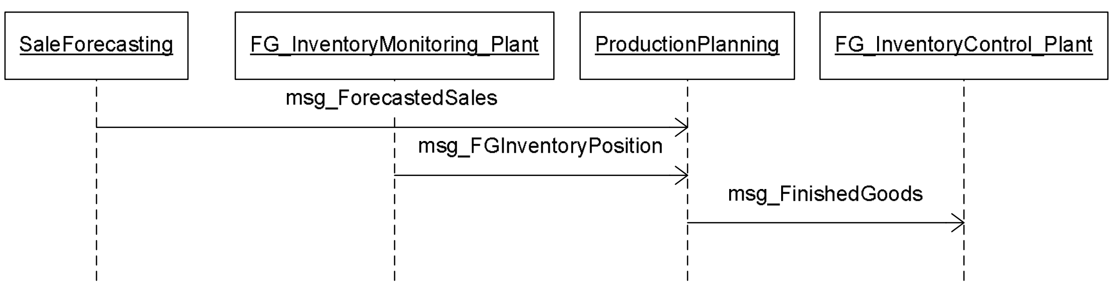

Figure B-2: Sequence diagram for production planning

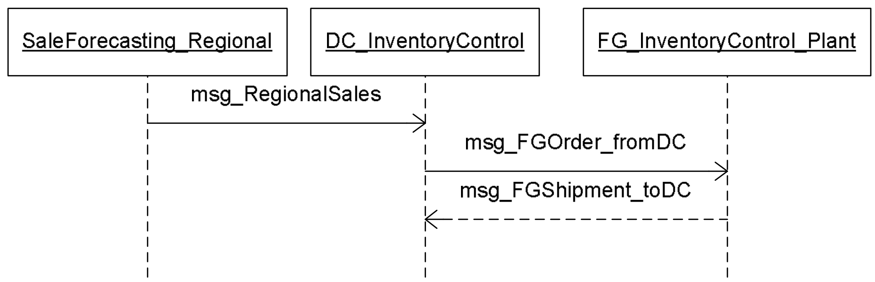

Figure B-3: Sequence diagram for distribution planning

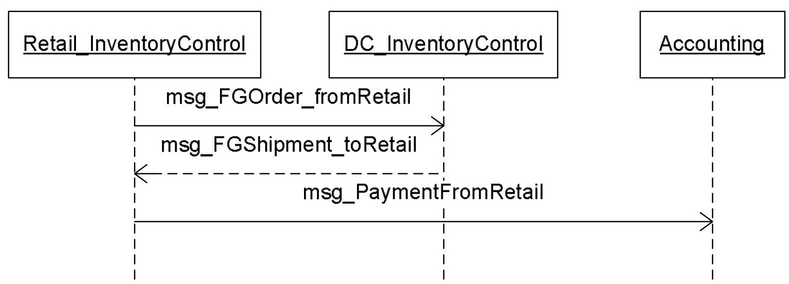

Figure B-4: Sequence diagram for product delivery

Appendix C

Class Diagrams for Supply Chain Roles

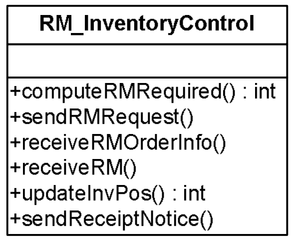

Figure C-1: Class diagram for the raw material inventory control role

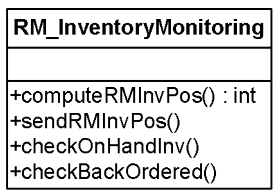

Figure C-2: Class diagram for the raw material inventory monitoring role

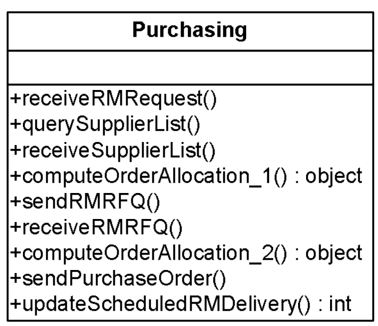

Figure C-3: Class diagram for the raw material purchasing role

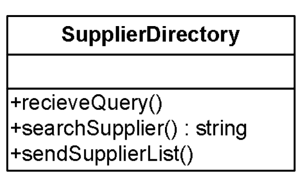

Figure C-4: Class diagram for the supplier directory role

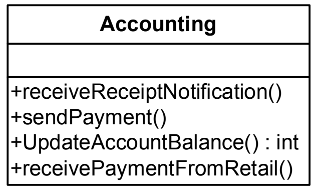

Figure C-5: Class diagram for the accounting role

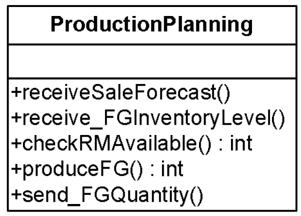

Figure C-6: Class diagram for the production planning role

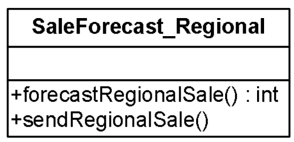

Figure C-7: Class diagram for the regional sale forecast role

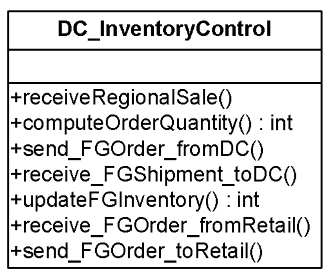

Figure C-8: Class diagram for the warehouse inventory control role

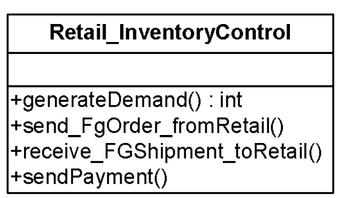

Figure C-9: Class diagram for the retailer inventory control role

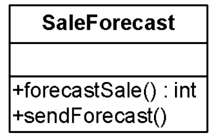

Figure C-10: Class diagram for the aggregated sale forecast role

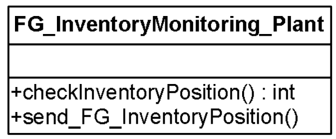

Figure C-11: Class diagram for the finished good inventory monitoring role at the plant

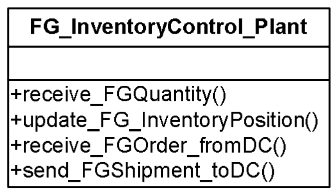

Figure C-12: Class diagram for the finished good inventory control role at the plant

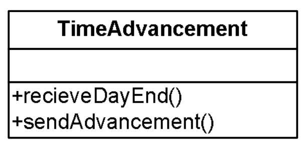

Figure C-13: Class diagram for the simulation time advancement role

Appendix D

Class Diagrams for Supply Chain Agents

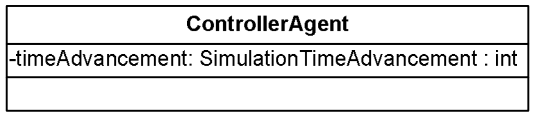

Figure D-1: Class diagram for the controller agent

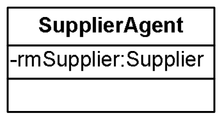

Figure D-2: Class diagram for the supplier agent

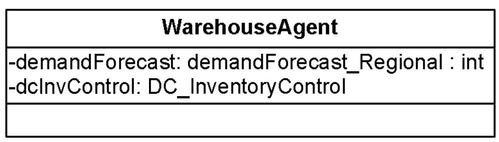

Figure D-3: Class diagram for the warehouse agent

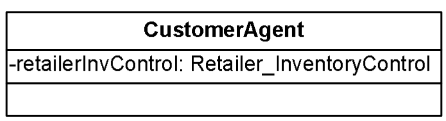

Figure D-4: Class diagram for the customer agent

Appendix E

Problem Settings for Supply Chain Network Redesign under Risk

Table E-1: Parameters and fixed costs for facilities for the medium problem

Fixed cost	Plant		Warehouse			
	no.1	**no.2**	**no.1**	**no.2**	**no.3**	**no.4**
Previously owned facility	yes	no	yes	no	no	no
Open cost	3000000	3500000	1000000	900000	1050000	850000
Rent cost	-	-	105000	85000	102000	88000
Close cost	450000	525000	150000	135000	157500	127500
Base operating cost	540000	630000	180000	162000	189000	153000
Reconfiguration budget	99999999					

Table E-2: Mean shipping costs for the medium problem

Shipping costs (s=supplier, p=plant, w=warehouse, c=customer)								
s	**p**	**cost/unit**	**p**	**w**	**cost/unit**	**w**	**c**	**cost/unit**
1	1	5	1	1	7	1	1	10
	2	4		2	5		2	9
2	1	4		3	7		3	8
	2	3		4	6		4	9
3	1	3	2	1	6		5	8
	2	3		2	5		6	7
4	1	5		3	8		7	10
	2	4		4	4		8	10
5	1	3				2	1	8
	2	4					2	8
6	1	5					3	9
	2	3					4	7
							5	9
							6	11
							7	10
							8	7
						3	1	6
							2	10
							3	7

	4	11
	5	9
	6	8
	7	10
	8	10
4	1	10
	2	11
	3	8
	4	8
	5	7
	6	9
	7	7
	8	6

Table E-3: Mean customer demands for the medium problem

Demand	**Units**
1	15000
2	13000
3	20000
4	16000
5	18000
6	12000
7	17000
8	11000

Table E-4: Mean supply quantities for the medium problem

Supply	**Units**
1	50000
2	43000
3	57000
4	40000
5	60000
6	42000

Table E-5: Mean facility capacities for the medium problem

Facility capacity			
plant	**Units**	**warehouse**	**Units**

1	100000	1	50000
2	180000	2	40000
		3	60000
		4	35000

Table E-6: Parameters and fixed costs for facilities for the large problem[2]

Fixed cost	Plant				Warehouse							
	no.1	no.2	no.3	no.4	no.1	no.2	no.3	no.4	no.5	no.6	no.7	no.8
Previously owned facility	no	yes	no	no	yes	no	no	yes	no	no	no	no
Open cost	3000	3500	2500	4000	1000	900	1050	850	650	1350	1200	905
Rent cost	-	-	-	-	105	85	102	88	64	140	130	90
Close cost	450	525	375	600	150	135	157.5	127.5	97.5	202.5	180	135.75
Base operating cost	540	630	450	720	180	162	189	153	117	243	216	162.9
Reconfiguration budget	99999.999											

Table E-7: Mean shipping costs for the large problem

Shipping costs (s=supplier, p=plant, w=warehouse, c=customer)																	
s	**p**	**cost**	**p**	**w**	**cost**	**w**	**c**	**cost**	**w**	**c**	**cost**	**w**	**c**	**cost**	**w**	**c**	**cost**
1	1	3	1	1	4	1	1	7	3	1	8	5	1	12	7	1	11
	2	3		2	5		2	7		2	8		2	12		2	10
	3	5		3	6		3	8		3	7		3	9		3	10
	4	4		4	7		4	9		4	9		4	8		4	9
2	1	2		5	8		5	10		5	10		5	9		5	10
	2	6		6	7		6	11		6	11		6	7		6	11
	3	5		7	6		7	6		7	11		7	9		7	12
	4	4		8	5		8	6		8	10		8	6		8	10
3	1	3	2	1	8		9	7		9	9		9	9		9	11
	2	4		2	7		10	11		10	8		10	10		10	9
	3	5		3	6		11	10		11	6		11	9		11	8
	4	3		4	5		12	8		12	9		12	9		12	9
4	1	2		5	4		13	9		13	8		13	9		13	9
	2	6		6	5		14	7		14	10		14	7		14	5
	3	6		7	6		15	8		15	11		15	6		15	11
	4	2		8	7		16	9		16	12		16	8		16	6
5	1	3	3	1	5	2	1	6	4	1	6	6	1	7	8	1	7
	2	4		2	4		2	6		2	5		2	5		2	7

[2] All costs shown in thousands

	3	3		3	6		3	6		3	5		3	12		3	6
	4	5		4	6		4	7		4	6		4	5		4	7
6	1	6		5	4		5	8		5	7		5	6		5	6
	2	2		6	7		6	9		6	10		6	6		6	6
	3	5		7	8		7	10		7	9		7	9		7	8
	4	4		8	4		8	9		8	12		8	8		8	9
7	1	4	4	1	7		9	8		9	12		9	9		9	10
	2	4		2	8		10	9		10	8		10	7		10	12
	3	4		3	7		11	6		11	7		11	10		11	11
	4	5		4	6		12	5		12	6		12	11		12	9
8	1	3		5	5		13	12		13	6		13	11		13	8
	2	3		6	7		14	11		14	8		14	9		14	7
	3	6		7	4		15	8		15	10		15	9		15	6
	4	2		8	6		16	8		16	10		16	9		16	5
9	1	6															
	2	6															
	3	2															
	4	5															
10	1	5															
	2	4															
	3	3															
	4	2															
11	1	3															
	2	4															
	3	5															
	4	6															
12	1	3															
	2	5															
	3	4															
	4	2															

Table E-8: Mean customer demands for the large problem

Demand	**Units**
1	15000
2	13000
3	20000
4	16000
5	18000
6	12000

7	17000
8	11000
9	10000
10	8000
11	10500
12	17250
13	9000
14	21000
15	23000
16	11500

Table E-9: Mean supply quantities for the large problem

Supply	**Units**
1	50000
2	43000
3	57000
4	40000
5	60000
6	42000
7	35000
8	45000
9	68000
10	61000
11	52000
12	55000

Table E-10: Mean facility capacities for the large problem

Facility capacity			
plant	**Units**	**warehouse**	**Units**
1	100000	1	50000
2	180000	2	40000
3	75000	3	60000
4	250000	4	35000
		5	33000
		6	68000
		7	60000
		8	42500

Appendix F

Problem Data for Assessment of the Reconfigurable Supply Chain

Table F-1: Parameters and fixed costs for facilities for the test problem

Fixed cost	**Plant**		**Warehouse**		
	no.6	**no.7**	**no.8**	**no.9**	**no.10**
Previously owned facility	no	no	no	no	no
Open cost	500000	400000	100000	90000	110000
Rent cost	-	-	105000	85000	114000
Close cost	750000	600000	150000	135000	165000
Base operating cost	900000	720000	180000	162000	198000
Reconfiguration budget	99999999				

Table F-2: Mean customer demands for the test problem

Demand	**Units**
1	15000
2	13000
3	20000
4	17000
5	18000
6	14000

Table F-3: Mean supply quantities for the test problem

Supply	**Units**
1	50000
2	43000
3	57000
4	52000
5	48000

Table F-4: Mean facility capacities for the test problem

Facility capacity			
plant	**Units**	**warehouse**	**Units**
1	100000	1	50000
2	100000	2	40000
		3	60000

VITA

Satama Sirivunnabood

Satama Sirivunnabood is a Ph.D. candidate in Industrial Engineering and Operations Research at Harold and Inge Marcus Department of Industrial and Manufacturing Engineering, the Pennsylvania State University. He holds a Bachelor's degree in mechanical engineering from King Mongkut's Institute of Technology Ladkrabang, Thailand in 2002 and the Master of Science in Advanced Manufacturing Technology and Systems Management from the School of Mechanical, Aerospace, and Civil Engineering, the University of Manchester (previously UMIST) in 2004. Prior to attending his Ph.D. program, Sirivunnabood had been working as a production engineer and a mechanical project engineer at the paper and packaging business of Siam Cement Group (Thailand). Apart from being a Ph.D. student, Sirivunnabood was working as a part-time IT specialist at the Richtsmeier lab, the Department of Anthropology while he had been at Penn State. Sirivunnabood is currently a student member of IEEE and INFORMS with special interests in information technology and network optimization in supply chains.