2005

A Methodology For Minimizing The Oscillations In Supply Chains Using System Dynamics And Genetic Algorithms

Ramamoorthy C.V.V. Lakkoju
University of Central Florida

A METHODOLOGY FOR MINIMIZING THE OSCILLATIONS

IN SUPPLY CHAINS USING

SYSTEM DYNAMICS AND GENETIC ALGORITHMS

By

RAMAMOORTHY C.V.V. LAKKOJU
B.Tech., S.V.U.C.E., Tirupati, India, 2002

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science
in the Department of Industrial Engineering and Management Systems
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer 2005

ABSTRACT

Supply Chain Management (SCM) is a critically significant strategy that enterprises depend on to meet challenges that they face because of highly competitive and dynamic business environments of today. Supply chain management involves the entire network of processes from procurement of raw materials/services/technologies to manufacturing or servicing intermediate products/services to converting them into final products or services and then distributing and retailing them till they reach final customers.

A supply chain network by nature is a large and complex, engineering and management system. Oscillations occurring in a supply chain because of internal and/or external influences and measures to be taken to mitigate/minimize those oscillations are a core concern in managing the supply chain and driving an organization towards a competitive advantage.

The objective of this thesis is to develop a methodology to minimize the oscillations occurring in a supply chain by making use of the techniques of System Dynamics (SD) and Genetic Algorithms (GAs). System dynamics is a very efficient tool to model large and complex systems in order to understand their complex, non-linear dynamic behavior. GAs are stochastic search algorithms, based on the mechanics of natural selection and natural genetics, used to search complex and non-linear search spaces where traditional techniques may be unsuitable.

ACKNOWLEDGEMENTS

This thesis was made possible with the help, cooperation and contribution of many people and I wish to wholeheartedly thank them all very much. First and foremost, I wish to thank and express my utmost gratitude to my advisor, Dr. Luis Rabelo, for his invaluable guidance, support and encouragement through out this research work. This thesis would not have been possible without the efforts and guidance of my advisor.

I wish to deeply thank my family, which has been highly instrumental in whatever I have accomplished till today.

I deeply thank all my friends for all their support, encouragement and good wishes wherever I go. I thank my best friend Deepak for all his help and support with the C++ programming. Also, I would like to thank my friends Ashish and Hamid for their help in getting me started with Vensim modeling and Genetic Algorithms respectively.

Finally, I would like to thank all other UCF staff and faculties who helped me in various ways throughout the duration of my studies.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

This chapter introduces and presents a brief overview of this research work. The objective behind this research work and the methodology employed to accomplish it are summarized here. This thesis presents a new methodology to mitigate the oscillations occurring in a supply chain. Various concepts from different fields have been blended together to accomplish this task. These concepts are introduced and discussed in this chapter and finally an outline of the thesis is presented.

1.1 Supply Chain Management

The management of the supply chain is one of the classical business problems. A supply chain is a network of facilities that procure raw materials/services/technologies, transform them into intermediate goods and final products/services, and deliver the products/services to customers through a distribution system. The purpose of supply chain management is to provide the right quantity of the right product at right time to the right customers at an optimal cost. A typical supply chain comprises of five elements: suppliers, manufacturers, distributors, retailers and customers. The process of integrating all these elements involves the coordination and cross-functioning of production planning, purchasing, material management, production, distribution, transportation, customer service and sales forecasting [1].

SCM has received much attention in academic and business circles because of its innovative approach to business [7]. The classical way of managing a supply chain was to observe and analyze the sales, demand and inventory values at the end of certain pre-defined time and fill the required gap in it. This methodology was based on the assumption that the supply and demand would remain linear and no drastic fluctuations would occur. Above that, this methodology was good for previous decades where supplier based market dominated the consumer-based market. However, with the increased competition, this supply-based market got replaced by consumer-based market where there were plenty of suppliers to satiate the consumers' demand. Above that, the push manufacturing concepts got replaced by pull manufacturing concepts and the importance of quality and service in time increased manifolds [2]. Added to these changes, as time passed by, with the latest advancements in information technology, corporations started using computerized systems to manage their supply chains. Enterprise Resource Planning (ERP), advanced web-based technologies and information systems have changed the way companies do their business and manage their supply chains. More and more attention is now focused on gathering real-time data and managing the supply chains through real-time networks.

The increasing competitive pressures in the global marketplace coupled with the rapid advances in information technology and the strategic policies that companies plan to adopt have made supply chains more and more complex, non-linear and dynamic in nature. The influence of internal and/or external causes on the non-linear dynamic nature produces oscillations in the supply chain. These oscillations, if not taken care of properly, could result in unwanted behavior and poor performance of the supply chain.

The objective of this thesis is to develop a methodology to minimize/mitigate the oscillations occurring in a supply chain by utilizing the techniques of system dynamics and genetic algorithms. These concepts are briefly discussed in the following sections and are explained in detail in the following chapters.

1.2 System Dynamics

The concept of System Dynamics (SD) was first developed by Jay W. Forrester in 1950's at the M.I.T., Cambridge, MA. Since then, it has been used as an efficient management tool for modeling complex real-world systems, understanding their behavior and implementing strategic policies [1]. Lertpattarapong [1] states that system dynamics is an efficient approach for exploring the non-linear dynamic behavior of a system and studying how the structure and the parameters of the system lead to behavior patterns.

The conceptual idea behind system dynamics is that all the objects in a system interact through causal relationships. These relationships come about through feedback loops, which control the interactions between the system objects. System dynamics asserts that these relationships form the underlying structure for any system. The creation of a complete dynamic model of a system requires the identification of reference modes and causal relationships that form the system's feedback loops. [5] Feedback system mentioned here refers to the scenario where variable A affects variable B and B in turn affects A through a series of cause and effects.

System dynamics is different from the other approaches in studying complex systems, mainly because of its extensive use of feedback loops. Stocks and Flows, which are the basic building blocks of system dynamics models, help describe how a system is connected by feedback loops. Once the model is created, computer software is used to simulate the model of the situation being studied. Performing "what-if" simulation analyses to test various policies on such a model greatly aids in understanding how the system changes over time [8].

System dynamics has been widely used to model complex supply chains, understand their behavior and design effective policies. The supply chain model used in this thesis work has been exclusively modeled using system dynamics concepts. The complete description of the model and its analysis are presented in Chapter 3.

1.3 Genetic Algorithms

Genetic Algorithms (GAs) are search and optimization procedures that are motivated by the principles of natural genetics and natural selection [43]. The concept of GAs was first developed during 70's by John Holland and his students at the University of Michigan, Ann Arbor [44]. The goals of their research have been twofold: (1) to abstract and rigorously explain the adaptive processes of natural systems, and (2) to design artificial systems software that retains the important mechanics natural selection [44]. Eventually, this approach has led to important discoveries and advancements in both natural and artificial systems science.

Over the last decade, GAs have been extensively used as search and optimization tools in various problem domains, including science, commerce and engineering [43]. GAs have been found very successful in arriving at an optimal/near-optimal solution to complex optimization problems, where traditional search techniques fail or converge to a local optimum solution. The primary reasons for their success are their broad applicability, ease of use and global perspective [44].

Koza [46] states that a Genetic Algorithm (GA) transforms a population (set) of individual objects, each with an associated fitness value, into a new generation of the population using the Darwinian principle of reproduction, survival of the fittest and analogs of naturally occurring genetic operations such as crossover and mutation. Each individual in the population represents a possible solution to a given problem. The genetic algorithm attempts to find a very good (or best) solution to the problem by genetically breeding the population of individuals over a series of generations.

In this research work, a genetic algorithm has been developed to minimize the oscillations occurring in the finished goods inventory of a supply chain model. The working principle of GAs, their advantages and the actual algorithm developed are explained in the following chapters.

1.4 Uniqueness and Contribution of this Thesis

Though many applications involving extensive use of system dynamics to model complex supply chains and genetic algorithms to solve various supply chain management problems were found in the literature, no application was found in the literature that combined both of these techniques, exploiting their combined power, towards improving the performance of a supply chain. The methodology proposed in this research work not only contributes to fill this gap but also provides a good field to delve in for further research.

In the actual methodology, a real-coded genetic algorithm (RCGA) has been developed and is integrated with a system dynamics supply chain model to minimize the oscillations occurring in the finished goods inventory. As part of the proposed methodology, a set of five important variables that are critical to the oscillations occurring in the finished goods inventory has been chosen. The RCGA then is used in conjunction with the system dynamics model to find out optimal/near-optimal values (for these five variables) that would minimize the oscillations.

The system dynamics model mentioned above has been built using a software called ‘Vensim 5.4’ and the real-coded genetic algorithm is developed using ‘C++’ language.

1.5 Thesis Outline

This thesis focuses on developing a methodology to minimize the oscillations occurring in a supply chain by utilizing genetic algorithms in conjunction with a system dynamics supply chain model. This thesis is organized as follows.

Chapter 1 serves as an introduction by presenting the problem to be solved and introducing the concepts that are used to solve the problem. Chapter 2 presents a literature review in the field of supply chain management and system dynamics, genetic algorithms and their applications in supply chain management. This chapter also presents the working principle of a conventional genetic algorithm. On the other hand, Chapter 3 presents the system dynamics modeling of the supply chain of a corporation called LSMC. The analysis and some important observations about the oscillations occurring in this model are also presented. After that, Chapter 4 describes the development of the real-coded genetic algorithm optimization module to minimize the oscillations in the finished goods inventory of LSMC’s supply chain. Finally, Chapter 5 presents the conclusions, scope and contribution of the thesis and ideas for potential future work.

CHAPTER 2: LITERATURE REVIEW

The scope of supply chain management (SCM) lies in its capacity of encompassing various business functions including but not limited to logistics, inventory management, demand forecasting, material and information flow, production planning and scheduling, and various other functions directly associated with enhancement of overall value for any business [2].

This research work contributes to the development of a methodology, which attempts to minimize the oscillations occurring in a supply chain by utilizing the tools of system dynamics (SD) and genetic algorithms (GA). System dynamics has been used here for modeling and understanding the behavioral dynamics of a supply chain and genetic algorithms to compute the near-optimal values of certain key parameters in that supply chain that would mitigate the oscillations occurring in it. A literature review has been done in order to understand the scope, various applications and theories underlying the concepts of system dynamics, genetic algorithms and their application towards supply chain management (SCM).

This literature review is divided into four sections, which basically encompass this entire research work. The first section discusses supply chain management and the application of various theories and analytical models for its continuous improvement. The second section discusses literature review in the field of system dynamics and its applications in supply chain management. The third and fourth sections present a literature review of genetic algorithms, their working principle and their various applications in the field of supply chain management.

Hence, all the research papers discussed in this chapter fall into one of these categories.

2.1 Supply Chain Management

Supply chain management, which is also emphasized by other similar terms such as value chain management, demand chain management, supply pipe-line management and network sourcing, is gaining more and more attention from researchers, academicians and business consultants due to its profound influence on the overall corporate performance [2]. Extensive research work is being carried out in the areas of demand forecasting, inventory management, logistics, transportation, information sharing and other areas, which contribute to better optimization of any supply chain. The objective of this literature survey is to briefly review the research in different facets of SCM, especially supply chain modeling, problems in SCM, inventory management, and supply chain optimization.

Croom *et al.* [29] examine various subject areas, which are considered core to the literature review of supply chain management. They provide a taxonomy of the field of supply chain management as an aid to both the classification of research in the field, and as a means of providing a framework for the identification of the key content of the subject. The authors classify the body of literature associated with SCM into different categories such as Purchasing and supply literature, Logistics and transportation literature, Marketing literature, Organizational behavior, industrial organization, transaction cost economics and contract view literature, Contingency theory, Institutional sociology, System engineering literature, Network literature, Best practices literature, Strategic management literature and Economic development literature. The authors also discuss how these subject areas have contributed to the overall SCM literature.

Tan [30] states that the advent of information technology and intense global competition has enticed many world-class manufacturers and service providers into adopting an integrated strategic approach to supply chain management. Using a survey of senior supply and materials management professionals in the US, the author investigates the contemporary practices and concerns of SCM. This study also relates the practices and concerns to firms' performance my means of bivariate correlation and multiple linear regression analysis and concludes that all of the significant SCM practices positively impact performance.

Croom and Giannakis [31] propose a conceptualization of the supply chain problem domain called the '3S Model'. The model highlights three dimensions of interest to supply chain scholars and practitioners, namely the *synthesis* of the business and resources network; the characteristics of *synergy* between different actors in the network; and the *synchronization* of all operational decisions related to the control of the production and delivery of goods and services.

Shapiro [32] present a review of the many challenges that supply chain modeling practitioners and their clients face when they set out to extend and apply strategic planning models that analyze wider and deeper decision problems. The author studies these challenges in the context of four categories of modeling and organizational imperatives, namely enlarging the scope of supply chain planning studies and models; reflecting theories of strategy in data-driven optimization models; formalizing scenario planning, applying stochastic programming and modeling risk; and expanding business processes to exploit fact-based analysis of strategic plans.

Talluri and Baker [34] propose a multi-phase mathematical programming approach for effective supply chain design. The proposed methodology develops and applies, in three phases, a combination of multi-criteria efficiency models, based on game theory concepts, and linear and integer programming methods. In phase I of the decision making process, multi-criteria efficiency models are utilized to evaluate the performance of suppliers, manufacturers and distributors. In phase II, an integer programming problem is utilized to design the SCN subject to efficiency, capacity, and location constraints. Phase III solves a transshipment problem in order to identify optimal routing decisions. The authors also detail the model application and insights through numerical illustrations.

Riddalls *et al.* [22] present a review of various mathematical methods used to model and analyze supply chains. The authors categorize these methods as continuous time differential equation models, discrete time difference equation models, discrete event simulation models and operational research techniques. They conclude that, while OR techniques are useful in providing solutions to local tactical problems, the impact of these solutions on the global behavior of the whole supply chain can only be assessed using dynamic simulation.

Terzi and Cavalieri [33] provide a comprehensive review made on more than 80 articles, with the main purpose of ascertaining which general objectives simulation is generally called to solve, which paradigms and simulation tools are more suitable, and deriving useful prescriptions both for practitioners and researchers on its applicability in decision-making processes within the supply chain context. The authors report that network SC design, SC strategic decision support, demand and sales planning, inventory planning, distribution and transportation planning, and production planning and scheduling are some of the important aspects of SCM where simulation has been applied successfully. Also, authors highlight the importance of discrete event simulation, parallel distributed simulation (PDS) and the high level architecture (HLA) in the context of SCM.

Agrawal *et al.* [35] consider the dynamic version of the problem of inventories to a set of retailers for rectifying the imbalance of inventories amongst them. The authors study the dynamic allocation problem with two decisions, the timing of the balancing shipments and determination of the new stocking levels at the retailers, as a dynamic program. They obtain structural properties for the optional allocation and timing strategies and present conditions under which a greedy heuristic to decide how much to ship from one retailer to another in optimal. They present an algorithm to solve the dynamic program efficiently and provide a heuristic solution procedure to the dynamic allocation policies.

Chiang and Monahan [36] present a two-echelon dual-channel inventory model based on queuing models. Using analytical methods, the authors develop operational measures of supply chain flexibility by defining a cost structure which captures two different operational cost factors: inventory holding cost and lost sales cost. To evaluate the possible benefits of using the dual-channel strategy, authors examine the performance of two other channel strategies: retail-only and direct-only strategies. Based on numerical examples, they conclude that the dual-channel strategy is dominant in most cases.

Giannoccaro et al. [37] address two key issues of inventory management in supply chains, namely the uncertainty associated with market demand and inventory related costs and the need of a tight integration among the supply chain stages. The authors propose a methodology to define a supply chain inventory management policy, which is based on the concept of echelon stock and fuzzy set theory. The echelon stock concept is adopted to manage the supply chain inventory in an integrated manner, whereas fuzzy set theory is used to properly model the uncertainty associated with both market demand and inventory costs. The proposed methodology is applied to a three stage supply chain. Further, authors quote that the use of fuzzy set theory is more appropriate than stochastic techniques to address uncertainty in market demand and inventory related costs, especially when the market is complex and turbulent.

Kapoor et al. [38] present a technical framework that supports sense-and-respond (SaR), an approach that enables enterprises to adapt to a rapidly changing business environment. To implement SaR approach, an enterprise proactively monitors trends and uses effective decision support tools to help it act in a timely manner. The proposed SaR approach combines the domain knowledge expertise in SCM, data warehousing, On-line analytical processing (OLAP) and J2EE Technologies. The authors describe two pilot projects in IBM where SaR approach has been implemented to solve business problems. The model improved the inventory management processes by diagnosing supply shortfalls, backlog accumulation, and inadequate inventory levels at the strategic stock points.

Dejonckheere [39] et al. propose a methodology based on control systems engineering that introduces a general decision rule for avoiding variance amplification (bullwhip effect), when the demand has to be forecasted. The authors first prove that whatever forecasting method is used (simple exponential smoothing, moving averages or demand signal processing), order-up-to replenishment systems will always result in the bullwhip effect. The authors state that the crucial difference with the class of order-up-to policies is that in the proposed rule, net stock and on order inventory discrepancies are only fractionally taken into account.

Jung *et al.* [40] present a simulation based optimization approach to SCM under demand uncertainty. The authors propose the use of deterministic planning and scheduling models which incorporate safety stock levels as a means of accommodating demand uncertainties in routine operation. The problem of determining the safety stock level to use to meet a desired level of customer satisfaction is addressed using a Monte-Carlo simulation based optimization approach. An industrial-scale case problem is presented to demonstrate the utility of the proposed approach.

Samaddar *et al.* [41] present a theoretical framework to investigate the relationships between the design of a supply network (SN) and inter-organizational information sharing (IIS). Theoretical arguments and analysis of secondary data are used to develop propositions regarding the association between SN configurations and IIS types, and the role of coordination structure in such associations.

Kelle and Akbulut [42] discuss the role of ERP tools in supply chain information sharing, cooperation and cost optimization. The authors concentrate on the inventory management aspects of supply chain coordination reviewing the recent quantitative modeling and organizational results available in literature. The authors present their analysis and discussion about: how to motivate companies for information sharing (by quantifying potential benefits and margins); how to select and aggregate data to share with partners; and how to extend the traditional vertically integrated business model.

2.2 System Dynamics and its applications in Supply Chain Management

The concept of System Dynamics (SD) was first developed by Jay W. Forrester in 1950's at the M.I.T., Cambridge, MA. Since then, it has been used as an efficient management tool for modeling complex real-world systems, understanding their behavior and implementing strategic policies [1]. Lertpattarapong [1] quotes that system dynamics is an efficient approach for exploring the non-linear dynamic behavior of a system and studying how the structure and the parameters of the system lead to behavior patterns. Another fundamental purpose of system dynamics is to design effective and robust policies, which enhance performance in managed systems.

Supply Chain Management (SCM) is one of the areas where system dynamics has been widely used to model complex supply chains, in order to understand their complex dynamic behavior. Angerhofer and Angelides [20] state that the application of system dynamics modeling to SCM has its roots in 'Industrial Dynamics', a field developed from the work of Jay Forrester at M.I.T. Towill, D. R. [21] regards Forrester as not only the father of System Dynamics but also as the originator of the many of the techniques of modern supply chain management.

This research work utilizes a supply chain model, built exclusively using the technique of SD, for the purpose of minimizing the oscillations occurring in that supply chain. The very first task involved in this process is to simulate this model over a period of three years to understand the behavior of various variables, stocks and flows involved in it. Hence, it becomes imperative to understand the technique of system dynamics and its applications in SCM. A literature survey has been done to understand how SD has been applied to modeling supply chains and the findings are discussed in this section. (The actual modeling procedure using system dynamics is presented in Chapter 3).

Angerhofer and Angelides [20] state that the current research in system dynamics modeling of supply chain management focuses on inventory decision and policy development, time compression, demand amplification, supply chain design and integration, and international supply chain management. Their paper gives an overview on the evolution of the research review in this field and also presents taxonomy of research and development in system dynamics modeling of supply chain management.

Riddalls *et al.* [22] present a review of various mathematical methods used to model and analyze supply chains, and appraise each method from a system dynamics perspective. The authors categorize these methods as continuous time differential equation models, discrete time difference equation models, discrete event simulation models and operational research techniques. They conclude that, while OR techniques are useful in providing solutions to local tactical problems, the impact of these solutions on the global behavior of the whole supply chain can only be assessed using dynamic simulation.

Rabelo *et al.* [5] propose a methodology that combines system dynamics and neural networks analysis for addressing the aspect of how enterprises can detect supply chain behavioral changes due to endogenous and /or exogenous influences and to predict such changes and their impacts in the short and long term horizons. The authors use system dynamics to model and analyze supply chain behavior and neural networks' pattern recognition abilities are then used to analyze simulation results and predict changes early before they take place. Also explained in this paper is how eigen-value analysis can be used to enhance the understanding of the problematic behavior.

Ovalle and Marquez [23] present a system dynamics based simulation study to assess the effectiveness of using e-collaboration tools in supply chain management. Their paper presents a classification of 'managerial spaces' where multiple trading partners share critical information using e-collaboration tools. After validating the output of the simulation model they state that it is certain that gradual increments of information sharing through collaborative forecasting and collaborative planning produce positive increases in the local and global performance of the SC.

Minegishi and Thiel [24] discuss how system dynamics can contribute towards improvement of knowledge for logistic behavior in an integrated food industry. They present the structure of a generic model and some simulation results as applied to the field of poultry production and processing. As an example of application, the authors present the consequences of dioxin infection on the supply chain of chicken industry.

Ge *et al.* [25] present a system dynamics approach for the analysis of the demand amplification problem (Bullwhip Effect). The authors utilize a system dynamics model of a part of a supermarket chain in the UK to investigate the causes of the dynamic behavior of the system and the sources of amplification from the downstream to the upstream of the chain. The major objective of this research is to investigate the impact of various information delays, demand forecasting and information sharing on the performance of the supply chain. The analysis reveals that information sharing is more important than the methods used in forecasting and the speed of information transmission.

Vlachos *et al.* [26] present the development of a dynamic SD-based model for strategic remanufacturing and collection capacity planning of a single product reverse supply chain for product recovery. The proposed approach enables development of efficient capacity planning policies for remanufacturing facilities in reverse supply chains, taking into account both economic and environmental issues.

Spengler and Schroter [27] modeled, using system dynamics, an integrated production and recovery system for supplying spare parts to evaluate possible strategies for meeting spare-parts demand for electronic equipment in the end-of-life service period. The authors use this model to examine whether and how component recovery contributes to supplying spare parts in the end-of-life service phase and reducing the costs of tack back and product recycling; to test alternative policies for managing spare parts in closed-loop supply chains and to examine the behavior of the system, especially when the planners underestimate demand for spare parts and decision makers must meet the customer requirements.

Rojas *et al.* [28] present a system dynamics methodology to understand the dynamics of officer supply chain in merchant marine in the UK by considering a part of the cadet training sub-system as an example. Their approach to manpower focuses on the flow of people and the flow of information through the manpower supply chain with the specific interest of different factors affecting this flow.

2.3 Genetic Algorithms

Genetic Algorithms (GAs) are search and optimization procedures that are motivated by the principles of natural genetics and natural selection [43]. The concept of GAs was first developed by John Holland and his students at the University of Michigan, Ann Arbor [44]. The goals of their research have been twofold: (1) to abstract and rigorously explain the adaptive processes of natural systems, and (2) to design artificial systems software that retains the important mechanics natural selection [44]. Eventually, this approach has led to important discoveries and advancements in both natural and artificial systems science.

Over the last decade, GAs have been extensively used as search and optimization tools in various problem domains, including science, commerce and engineering [43]. The primary reasons for their success are their broad applicability, ease of use and global perspective [44].

Koza [46] states that a Genetic Algorithm (GA) transforms a population (set) of individual objects, each with an associated fitness value, into a new generation of the population using the Darwinian principle of reproduction, survival of the fittest and analogs of naturally occurring genetic operations such as crossover and mutation. Each individual in the population represents a possible solution to a given problem. The genetic algorithm attempts to find a very good (or best) solution to the problem by genetically breeding the population of individuals over a series of generations.

2.3.1 Working Principle of Genetic Algorithms

There are different types of genetic algorithms, but fundamentally they can be classified into two types, namely Binary Coded GAs and Real Coded GAs (RCGAs). The working principle of a conventional/binary coded algorithm is presented in this section. Other types of GAs work pretty much on the same principles. A simple GA that yields good results in many practical problems is composed of three operators: 1. Reproduction, 2. Crossover, and 3. Mutation. [44]

The description of conventional GA presented below is taken as it is from Koza's [46] paper on survey of genetic algorithms and genetic programming.

Before applying a GA to a problem, the user designs an artificial chromosome of a certain fixed size and then defines a mapping (encoding) between the points in the search space of the problem and instances of the artificial chromosome. For example, in applying the GA to a multidimensional optimization problem (where the goal is to find the global optimum of an unknown multidimensional function), the artificial chromosome may be a linear character string (modeled directly after the linear string of information found in DNA). A specific location (a gene) along this artificial chromosome is associated with each of the variables of the problem. Character(s) appearing at a particular location along the chromosome denote the value of a particular variable (i.e., the gene value or allele). Each individual in the population has a fitness value (which, for a multidimensional optimization problem, is the value of the unknown function). The genetic algorithm then manipulates a population of such artificial chromosomes (usually starting from a randomly-created initial population of strings) using the operations of reproduction, crossover, and mutation. Individuals are probabilistically selected to participate in these genetic operations based on their fitness. The goal of the genetic algorithm in a multidimensional optimization problem is to find an artificial chromosome which, when decoded and mapped back into the search space of the problem, corresponds to a globally optimum (or near-optimum) point in the original search space of the problem.

In preparing to use the conventional genetic algorithm operating on fixed-length character strings to solve a problem, the user must (1) determine the representation scheme, (2) determine the fitness measure, (3) determine the parameters and variables for controlling the algorithm, and (4) determine a way of designating the result and a criterion for terminating a run.

In the conventional genetic algorithm, the individuals in the population are usually fixed-length character strings patterned after chromosome strings. Thus, specification of the *representation scheme* in the conventional genetic algorithm starts with a selection of the string length *L* and the alphabet size *K.* Often the alphabet is binary, so *K* equals 2. The most important part of the representation scheme is the mapping that expresses each possible point in the search space of the problem as a fixed-length character string (i.e., as a *chromosome)* and each chromosome as a point in the search space of the problem. Selecting a representation scheme that facilitates solution of the problem by the genetic algorithm often requires considerable insight into the problem and good judgment.

The evolutionary process is driven by the *fitness measure*. The fitness measure assigns a fitness value to each possible fixed-length character string in the population.

The primary parameters for controlling the genetic algorithm are the population size, *M,* and the maximum number of generations to be run, *G.* Populations can consist of hundreds, thousands, tens of thousands or more individuals. There can be dozens, hundreds, thousands, or more generations in a run of the genetic algorithm.

Each run of the genetic algorithm requires specification of a *termination criterion* for deciding when to terminate a run and a method of *result designation.* One frequently used method of result designation for a run of the genetic algorithm is to designate the best individual obtained in any generation of the population during the run (i.e., the *best-*

so-far individual) as the result of the run. Once the four preparatory steps for setting up the genetic algorithm have been completed, the genetic algorithm can be run.

The evolutionary process described above indicates how a globally optimum combination of alleles (gene values) within a fixed-size chromosome can be evolved.

The three steps in executing the genetic algorithm operating on fixed-length character strings are as follows:

1. Randomly create an initial population of individual fixed length character strings.
2. Iteratively perform the following sub-steps on the population of strings until the termination criterion has been satisfied:
   (A.) Assign a fitness value to each individual in the population using the fitness measure.
   (B.) Create a new population of strings by applying the following three genetic operations. The genetic operations are applied to individual string(s) in the population chosen with a probability based on fitness.
      (i.) Reproduce an existing individual string by copying it into the new population.
      (ii.) Create two new strings from two existing strings by genetically recombining substrings using the crossover operation (described in the following section) at a randomly chosen crossover point.
      (iii.) Create a new string from an existing string by randomly mutating the character at one randomly chosen position in the string.

3. The string that is identified by the method of result designation (e.g., the best-so-far individual) is designated as the result of the genetic algorithm for the run. This result may represent a solution (or an approximate solution) to the problem.

2.3.2. Genetic Operators

As stated earlier, genetic algorithms consist of three operators, namely Selection or Reproduction Operator, Crossover Operator, and Mutation Operator.

Selection or Reproduction Operator: The genetic operation of *reproduction* is based on the Darwinian principle of reproduction and survival of the fittest. In the reproduction operation, an individual is probabilistically selected from the population based on its fitness (with reselection allowed) and then the individual is copied, without change, into the next generation of the population. The selection is done in such a way that the better an individual's fitness, the more likely it is to be selected. An important aspect of this probabilistic selection is that every individual, however poor its fitness, has some probability of selection.

There exist a number of ways to accomplish the above mentioned task. Some common methods are tournament selection, proportionate selection and ranking selection. Deb [43] states that the tournament selection has better or equivalent convergence and computational time complexity properties when compared to any other reproduction operator that exists in the literature.

Crossover Operator: The genetic operation of *crossover* (sexual *recombination)* allows new individuals (i.e., new points in the search space) to be created and tested. The operation of crossover starts with two parents picked randomly from the population and some portion of the strings are exchanged between the strings to create two new strings. Each offspring contains some genetic material from each of its parents.

Suppose that the crossover operation is to be applied to the two parental strings 10110 and 01101 of length L = 5 over an alphabet of size K = 2. The crossover operation begins by randomly selecting a number between 1 and (L-1) using a uniform probability distribution. Suppose that the third interstitial location is selected. This location becomes the crossover point. Each parent is then split at this crossover point into a crossover fragment and a remainder. The crossover operation then recombines remainder of string 1 (i.e., - - - 1 0) with crossover fragment 2 (i.e., 011 - -) to create offspring 2 (i.e., 01110). The crossover operation similarly recombines remainder 2 (i.e., - - - 01) with crossover fragment 1 (i.e., 101 - -) to create offspring 1 (i.e., 10101).

There are different types of crossover operators depending on whether the GA under consideration is binary coded or real coded. One point, n-point cross over are widely used in binary coded GAs. For real coded GAs there exist Blend Crossover (BLX), Simulated Binary Crossover (SBX), Linear Crossover, Simplex Crossover, etc.

Mutation Operator: The crossover operator is mainly responsible for search aspect of GAs, even though the mutation operator is also use for this purpose [43]. The need for mutation is to keep diversity in population. The operation mutation allows new individuals to be created. It begins by selecting an individual from the population based on its fitness (with reselection allowed). A point along the string is selected at random and the character at that point is randomly changed. The bitwise mutation operator changes a 1 to a 0, vice versa, with a certain mutation probability. The altered individual is then copied into the next generation of the population.

For of real parameter GAs, there exist mutation operators such as Random Mutation, Non-uniform Mutation, Polynomial Mutation, Normally Distributed Mutation, etc.

2.3.3. Differences between GAs and other traditional optimization techniques

Hongwei Ding *et al.* [17] presented some important differences between genetic algorithms and traditional optimization methods/algorithms. The following is a summary.

1. GAs use an encoding of the control variables, rather than the variables themselves.
2. GAs search from one population of solutions to another, rather than from individual to individual. It is a great advantage for searching noisy spaces littered with local optimum, instead of relying on a single configuration to search through the space.
3. GAs use only objective function information to guide itself through the solution space, not derivatives. Once GA knows the value of 'goodness' about a configuration, it can use this to continuing to approach the optimum.
4. GA is probabilistic in nature, not deterministic. This is a direct result of the randomization techniques used by GAs. It is not the case for most existing methods.
5. One of the most attractive advantages of using GA as a design tool is its ability to find solutions to problems in a way completely free of preconceptions about what is possible and what is not. This is something that human designers find very difficult.

2.4 Genetic Algorithms and their applications in Supply Chain Management

A literature review has been done to examine various applications of GAs in the field of SCM. Another important focus of this survey is to review the literature in SCM for applications that integrated genetic algorithms and system dynamics.

Sudhir and Rajendran [9] propose a genetic algorithm (GA) to optimize the base-stock levels of a single-product serial supply chain with the objective of minimizing the sum of holding and shortage costs in the entire supply chain. Simulation is used to evaluate the base stock levels generated by the GA. They prove the effectiveness of the proposed GA by comparing the optimal base-stock levels obtained through complete enumeration of the solution space with those yielded by the GA. The effectiveness of the proposed GA in terms of generating base-stock levels with minimum total cost is also compared with that of a random search procedure.

Pal *et al.* [14] propose a real-coded genetic algorithm (RCGA) to solve a constrained non-linear mixed integer program in a two-warehouse inventory control problem. The proposed RCGA, applicable for mixed variables (integer and non-integer), makes use of ranking selection, whole arithmetic crossover and mutation (uniform mutation for integer variable and non-uniform for non-integer variable). Elmahi *et al.* [18] propose a similar algorithm, but utilizing different genetic operators such as roulette wheel selection and R-point crossover, to optimize the scheduling of transport equipments of a supply chain under the just-in-time setting.

Rezg *et al.* [19] propose a methodology that combines a simulation model and a genetic algorithm (GA) for joint optimization of preventive maintenance and inventory control in a production line composed of *n* machines, in order to improve the productivity of the system. A simulation model built in Promodel is used to simulate the behavior of the production line of *n* machines under various maintenance and inventory control strategies where as the proposed binary coded GA optimizes parameters of the simulation model in each iteration. The authors, comparing the results of simulation with analytical results, state that the joint optimization method significantly reduced the total cost.

Truong, T.H. and Azadivar F [10] propose a hybrid approach that combines GAs, Mixed Integer Programming (MIP) and simulation techniques into one framework for designing an optimal configuration for a supply chain. Here, each configuration of supply chain is defined by a set of quantitative parameters (e.g. inventory levels, frequency of ordering etc.) as well as a set of policy and qualitative parameters (e.g. location of distribution centers, mode of transportation etc). In the proposed methodology, within a given iteration of the GA's operators, such as crossover and mutation, a new combination of values for the qualitative variables is created. The MIP is then used to solve for the corresponding quantitative variables. After all the variables are determined, a supply chain simulator is invoked that automatically simulates and evaluates the performance of the new configuration simulation model returns the overall long run system-wide cost and customer service level of the supply chain.

Tong Wu and Peter O'Grady [11] developed a methodology to improve the design and performance of a supply chain. A network-based approach, called Extended Trans-nets, has been presented which represents the design of the supply chain as an abstract network with 'AND' and 'OR' nodes. (Trans-Nets provide a method for abstracting the information in a network for supply chain modeling). They propose a constraint based genetic algorithm (CBGA) that is used as part of the Extended Trans-Net approach to search for improvements in the design that satisfy the constraints imposed on the system. This approach is unique in that the CBGA is designed to operate within the constraints inherent in supply chain design problems.

Cochran and Chen [12] propose a GA approach that dynamically optimizes multi-objective production planning decisions in a manufacturing environment on a daily basis. The actual production planning problem is formulated as a multi-objective optimization problem where the objective function is a combination of three single-objective functions. The proposed GA searches for best or optimal weight assignments and solves for the best daily production planning decisions. The population of chromosomes for the GA is generated such that the total weights on the three individual-objective production plans must sum to unity.

Felix T.S. Chan and S.H. Chung [13] propose a multi-criterion genetic algorithm in combination with analytic hierarchy process (AHP) to solve an order distribution optimization problem in a demand driven supply chain network. The problem attempted to solve here is to determine, for a group of collaborated suppliers, an optimal solution which indicates the allocation schedule of customers' orders to suitable suppliers. The uniqueness of this approach is that it utilizes AHP to evaluate the fitness values of the chromosomes.

Hongwei Ding *et al.* [15, 16, 17] have developed, utilizing genetic algorithms, simulation-based optimization methods for solving various supply chain management problems. In [15], they propose a simulation-based multi-objective optimization method for joint decision-making on strategic sourcing and inventory replenishment. This method enables decision-makers simultaneously optimize decisions at both strategic and operational levels. In this method, for every individual in the population the GA optimizer generates a set of possible values for decision variables using which, a discrete-event simulator simulates the corresponding configuration of the supply chain and yields an estimate of the costs and demand fill-rate which in turn are used by the GA for fitness evaluation and further processing. Based on a similar idea, the authors present, in [16], a methodology to design a production-distribution network. Here, a multi-objective GA based on non-dominated sorting is used to direct the search for compromised solutions under various conflicting criteria. Also, using almost a same methodology they propose another algorithm [17] for supplier selection problem.

2.5 Summary

This literature review focuses on the classical as well as the latest research ideas in the field of supply chain management. The review of applications of system dynamics in supply chain management proves that the concept of modeling supply chains using the technique of system dynamics has been exploited by many researchers and a wide range of problems in SCM have been modeled in system dynamics. The review of applications of genetic algorithms in SCM reveals that GAs have been used to solve a wide variety of complex problems in this field.

Inspite of the popularity of system dynamics for effectively modeling supply chains and the capability of genetic algorithms to provide solutions to a wide variety of complex problems, no application in literature was found that combined both of these techniques. This research work lays a foundation towards developing a methodology that combines system dynamics and genetic algorithms with the objective of mitigating the oscillations occurring in a supply chain. With further development this methodology has a strong potential to become an important tool in this field.

CHAPTER 3: SUPPLY CHAIN MODEL DESCRIPTION

A supply chain is a complex system by definition. In order to understand and explore the complexities of any supply chain, the dynamics underlying it has to be modeled using appropriate modeling tools [2]. As seen from the literature review, system dynamics is considered to be a very efficient tool for modeling a supply chain, as it is capable of capturing the non-linearities underlying a supply chain.

The aim of this thesis is not to develop a supply chain model, but is to develop an optimization module that would minimize the oscillations of an existing supply chain model. Hence, a comprehensive supply chain model has been utilized to study and minimize the oscillations occurring in it. This chapter briefly describes the methodology behind the construction of the supply chain model whose oscillations this thesis is attempting to minimize. This model has been adapted from Lertpattarapong's thesis [1] which was done in 2002 at M.I.T. It is about a leading semiconductor manufacturing corporation called LSMC (the actual name is omitted to respect confidentiality) which produces technological gadgets and complimentary products for personal computers. The company was facing the problem of having persistent oscillations in its finished goods inventory and desired capacity. Lertpattarapong [1] addressed this problem from the perspective of system dynamics, and after modeling the entire supply chain, he proved that these problems were totally endogenous and not exogenous as thought by the professionals of LSMC.

3.1 Modeling in System Dynamics

System Dynamics is a methodology for studying the dynamics of real-world systems. System Dynamics methodology has been utilized in modeling the supply chain of LSMC. The conceptual idea behind system dynamics is that all the objects in a system interact through causal relationships. These relationships come about through feedback loops, which control the interactions between the system objects. System dynamics asserts that these relationships form the underlying structure for any system. The creation of a complete dynamic model of a system requires the identification of reference modes and causal relationships that form the system's feedback loops. [5]

Reference modes are graphs that represent historical and projected behavior of certain variables of interest. It is not necessary that these graphs are to be drawn to a high degree of precision, but their trends and varying behaviors over time, such as increasing, decreasing, or oscillating, must be evident.

Feedback loops can be either negative or positive. A negative feedback loop is a series of causal relationships that tend to move behavior towards a goal. In contrast, a positive feedback loop is self-reinforcing; it amplifies disturbances in the system to create high variations in behavior.

Causal loop diagrams are important tools for representing the feedback structures of systems. A causal loop diagram consists of variables connected by arrows denoting the causal influence among the variables. Figure 3.1 depicts the types of causal relationships and the structure of a causal loop. [5]

Link polarity positive:
If Market Share increases (decreases),
then Demand increases (decreases)

Link polarity negative:
If Deliveries increase (decrease),
then Inventories decrease (increase)

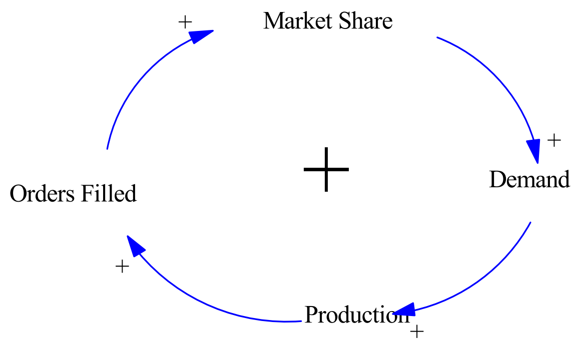


Figure 3.1 Causal loop diagrams

From causal loops, stock and flow structures can be developed. Stocks are accumulators of information or materials that characterize the state of the system. They generate the information upon which decisions and actions are based. They also create delays by accumulating the difference between the inflow and outflow of a process. Flows are rates that are added to (inflows) or subtracted from (outflows) a stock. Figure 3.2 shows the components of the stock and flow diagrams. Stocks are represented by blocks while flows by valves. This graphical description of the system based on stocks and flows can be mapped into a mathematical (differential, integral etc.) description of the system. [5]

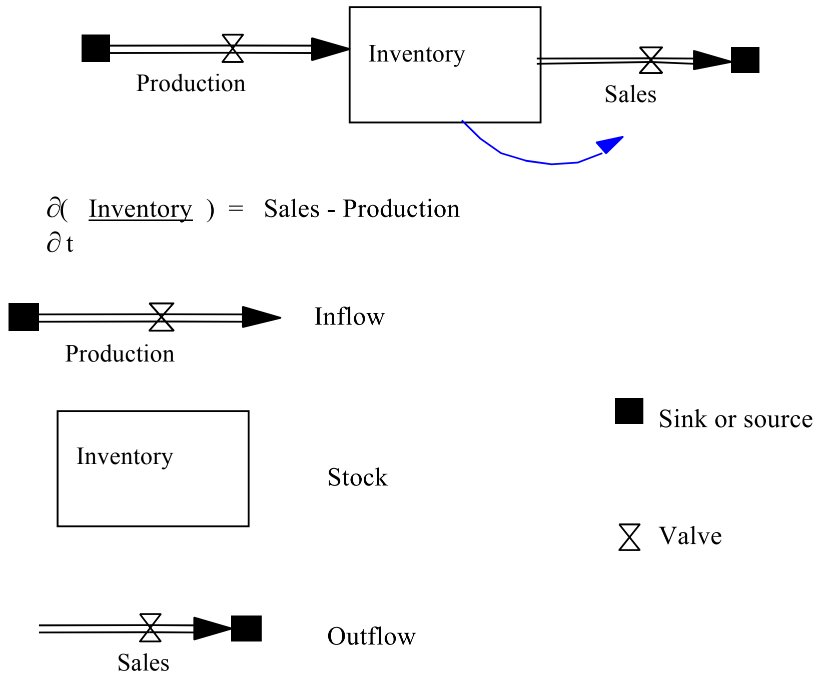


Figure 3.2 Stocks and flows in system dynamics

3.2 System Dynamics Model of LSMC's Supply Chain

The supply chain of any manufacturing industry consists of suppliers, manufacturers, distributors, retailers and customers as its five core elements. [1] The classic definition of a supply chain states that it is a network, which performs the operations of procuring the raw material from suppliers, transformation of this material into intermediate and finished goods by the manufacturer, and finally distributing these products to the customers through a chain of retailers. It is very important to integrate all this five elements in order to obtain an integrated supply chain. This integrated supply chain would increase the overall value of organization, business and the shareholders. [2]

As mentioned earlier, LSMC produces technological gadgets and complimentary products for personal computers (PCs) and hence its sales are directly related to the growth of the PC market, which has been very strong during the 1990s. Even though the company has maintained its market share, it experienced competitive pressures and demand fluctuations that have impacted its SC performance. The company supplies its products to Original Equipment Manufacturers (OEMs) like Dell, Gateway, and Hewlett-Packard. Since 1998, led by Dell, many OEMs changed their strategies by aggressively eliminating slack from their systems through the adoption of Build-To-Order (BTO) and Just-In-Time (JIT) processes that drastically reduced their inventory levels and allowed operating on only seven days or less of supply inventory. The OEMs' "slack" was basically shifted to LSMC, and the other suppliers in the PC supply chain, who faced even greater pressure because of that. Moreover, because the PC market is unpredictable LSMC occasionally was not able to keep up with the demand. Competitors, on the other hand, introduced greater variety of high performance, lower cost products. So, LSMC had to introduce more improved products in order to protect its existing and potential market share, and this has exacerbated its supply chain problems.

To address the company's supply chain problems a system dynamics model was developed. In order to capture all the relevant variables and underlying concepts and develop the final model, Lertpattarapong [1] interviewed various participants from different departments of LSMC. Participants included a senior manager from information technology department who was specialized in the company's SC, a manager from manufacturing department, a manager from a strategic planning department, two managers from SC department, and also engineers and scheduling planners from those departments. In addition, an analysis of historical data was conducted. Finally, reference modes for selected supply chain variables were developed in cooperation with the managers. The major parameters found during the analysis are explained in the following section.

3.2.1 Reference Modes for LSMC Supply Chain Modeling

1. *Product Life Cycle and Demand:* Product lifecycle has been decreasing for the new generations of products while demand on these new generations has been increasing. To retain its position as a market leader, LSMC is always under pressure to come out with new generations of its products. This drastically reduces its product life cycle. The hope is that the product life cycle would not decrease further with time.
2. *Actual Capacity Relative to Desired Capacity:* It was found that actual capacity relative to the desired capacity oscillates and the amplitude of oscillation is growing, indicating high instability in the supply chain.
3. *Change in Customer Orders:* Because of the Build to Order (BTO) policies of Original Equipment Manufacturers (OEM), the contracted cancellation periods have been reduced from 60 days to one week. This created highly unstable signals of customer orders in the supply chain.
4. *Raw Material Inventory Write-Off:* Due to high pace of customer order changes and short product life cycle, LSMC experiences increasing raw material inventory write-offs.
5. *Average OEM Margin:* OEMs have started introducing cheap products for low-end users who do not need powerful equipment. So LSMC had to introduce low-end products to retain its market share. The average profit margin has been decreasing over time.

6. *Pre-Assembly Component Inventory:* The pre-assembly component inventory is bought from various suppliers as well as manufactured indigenously at LSMC. LSMC managed to keep this inventory at minimum level in order to reduce the holding costs. However, because of the BTO policy and the complexity in the product configurations and architectures, it is expected that this inventory will start increasing in the near future.
7. *Throughput Time of Product Life Cycle Time and Working in Process (WIP) Inventory:* Due to the increasing complexity of LSMC's products and BTO policy of OEM's, the throughput time of product life cycle time and its working in process inventory of LSMC and its suppliers has been steadily increasing and that of OEM has been decreasing.
8. *OEM's Inventory:* Because of the Build to Order policies of OEM's, their inventories have reduced drastically, but it has resulted in higher inventories for LSMC and its suppliers.
9. *Product Inventory:* If demand is less than expected, LSMC will have to hold more finished goods inventory which results in extra holding costs and product obsolescence. But if it reduces its inventory and demand increases, then it will loose market share and potential revenues. Therefore, LSMC should reduce the mismatch between market demand and inventory level.

Once the reference modes are developed, the next step involved in the modeling process is to develop causal loop diagrams. The following section briefly describes the methodology behind the development of causal loop diagrams for LSMC supply chain modeling.

3.2.2 Causal Loop Diagrams

Causal loop diagrams are the basis on which the whole model is built. They depict, graphically, the interactions and cause-and-effect relationships among all model variables. For LSMC's supply chain, the causal loop diagram consists of seven segmental loops which are described as follows.

1. *LSMC's market share depends on its production capacity:* As capacity increases more shipments are made. This would increase LSMC's market share, which would further induce demand growth. If LSMC has lower capacity it will not be able to fill in orders, so customers might shift to competitors. This effect is depicted in Figure 3.3.

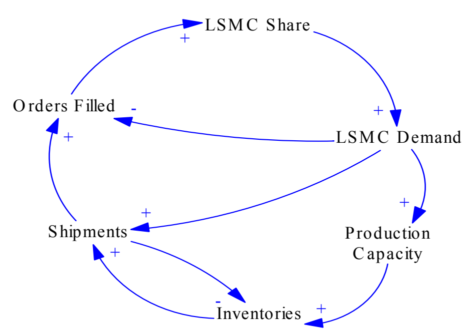


Figure 3.3 Increase in production will result in increase in demand

2. *LSMC's expansion depends on its market share:* As market share increases, more revenues are realized, and hence more money is invested in production capacity to meet the growing demand. This concept is reflected in Figure 3.4
3. *The competition for LSMC increases with the increase in profit and decreases with the decrease in profit:* As revenues increases profits increase. This induces competitors to enter the market, which reduces LSMC's market share and

consequently its profits. When profit margin decreases, and market becomes saturated, fewer competitors tend to enter this market. This logic is depicted in Figure 3.5

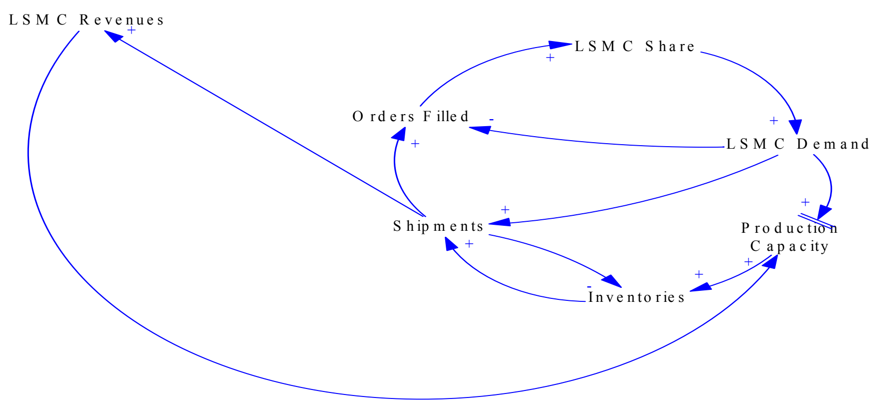


Figure 3.4 More market share, more expansion for LSMC

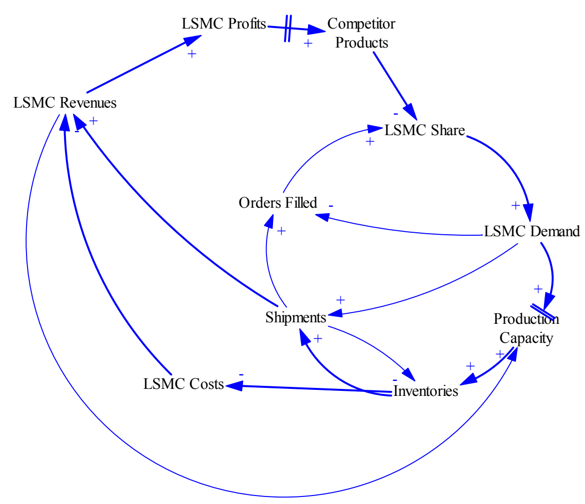


Figure 3.5 Increase in profit increases competition, and decrease in profit decreases competition

4. *Growth of LSMC:* As production capacity increases, it produces and ships more goods. The more the revenues the more the money is invested into R&D and better products are developed. Because of that LSMC has been able to sell products at a premium price compared to its competitors. This further fuels the growth of LSMC. This concept is reflected in Figure 3.6

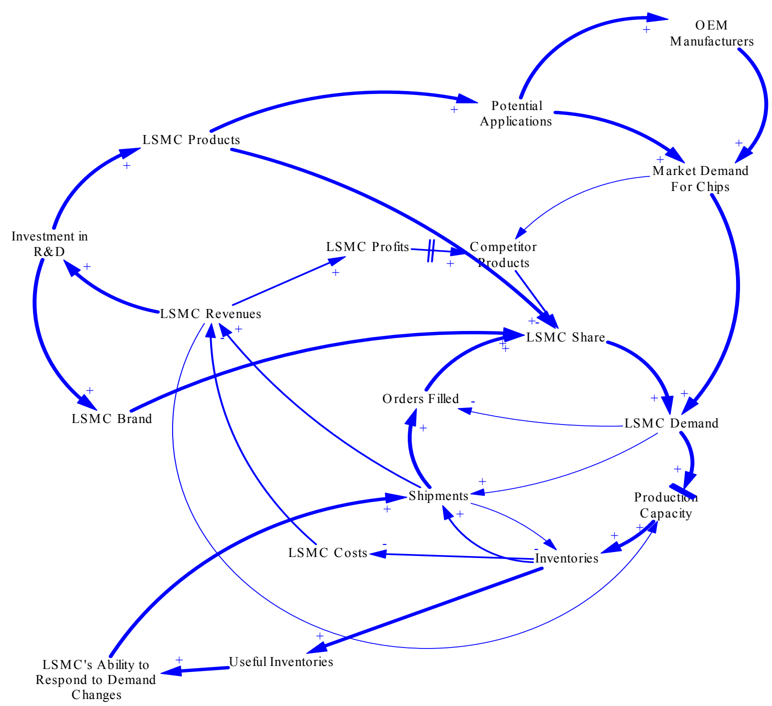


Figure 3.6 Growth of LSMC

5. *Product Life Cycle of LSMC Products:* LSMC tries to push its new products into market in order to beat competition. This results in the obsolescence of its old products. To sell off old products, it has to introduce huge price discounts, which reduces its potential revenues. Further, because of product obsolescence, the pre ordered material has to be written off which further contributes towards decreasing revenues. As product life cycle decreases, the ability to respond to orders decreases. This effect is shown in Figure 3.7

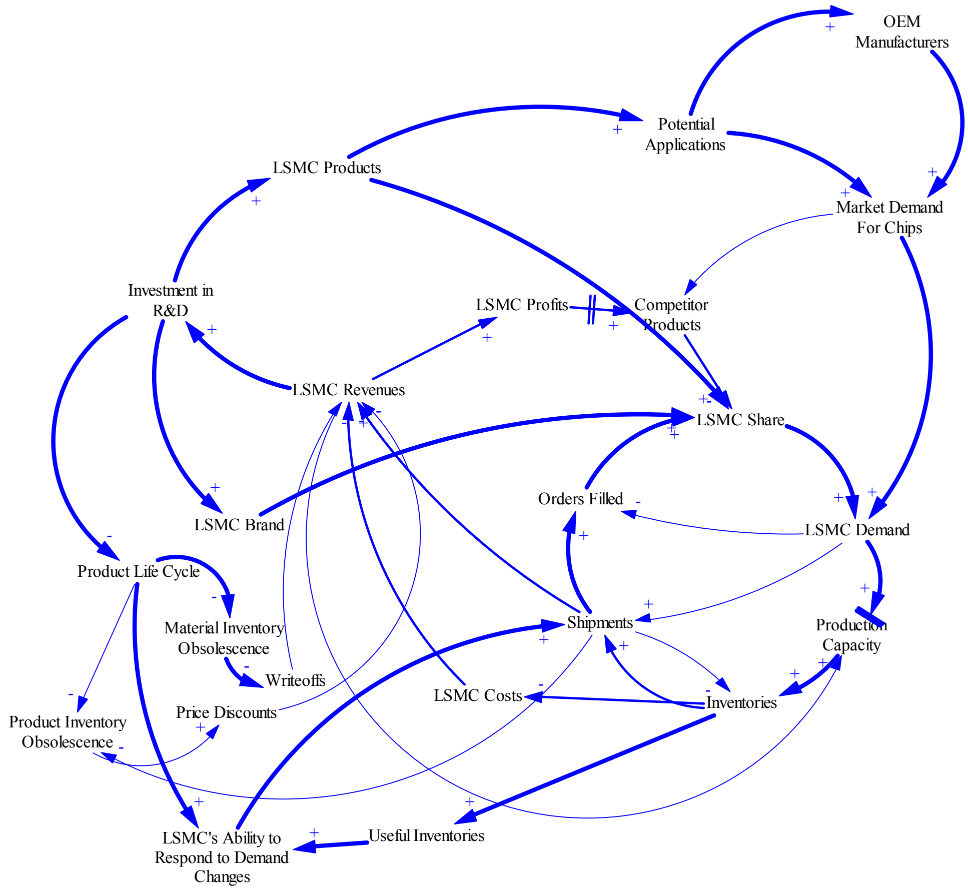


Figure 3.7 Product life cycle of LSMC products

6. *<u>Impact of Competition on LSMC Market Share:</u>* The growth in PC market increases competition in the OEM market. Due to this, intensive cost cuttings are going on in the OEM industry. OEMs work to reduce their inventories. To do so, they put pressure on suppliers like LSMC to decrease the cancellation time of their orders. Further, due to the growth in PC market, the computer chip market has been growing steadily too. This has resulted into the entry of many competitors of LSMC into the market. This effect is presented in Figure 3.8

7. *<u>Segmentation of the Market:</u>* Due to the decreasing product life cycle, LSMC has divided its market into high-end segment and low-end segment. This has helped

reduce the inventory write-off as well as loss due to high discounts that should be given. This policy has helped LSMC to increase its product life cycle. The problem is that capacity flexibility is reduced.

Now, all the above segmental loops can be combined together to form a complete causal loop diagram of LSMC supply chain. The complete causal loop diagram is shown in the Figure 3.9. The diagram is categorized in several areas such that it is easier to identify which areas in the SC system can contribute more to improving the SC behavior.

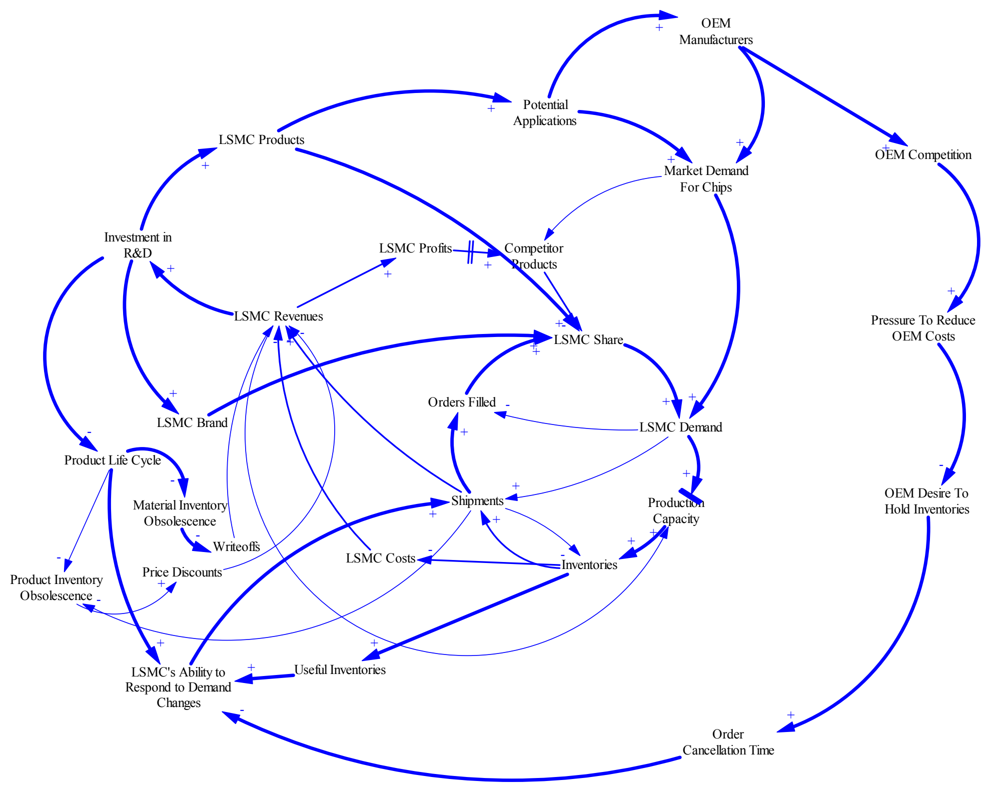


Figure 3.8 Impact of competition on LSMC market share

Market and OEM areas are obviously external and not under the control of LSMC. As will be mentioned later, the analysis using the system dynamics model showed that the production management policies of LSMC were the direct causes of the oscillatory behavior. In other words, the problems were due to internal factors/policies although they were triggered by external factors in the OEM and Market segments of the supply chain system.

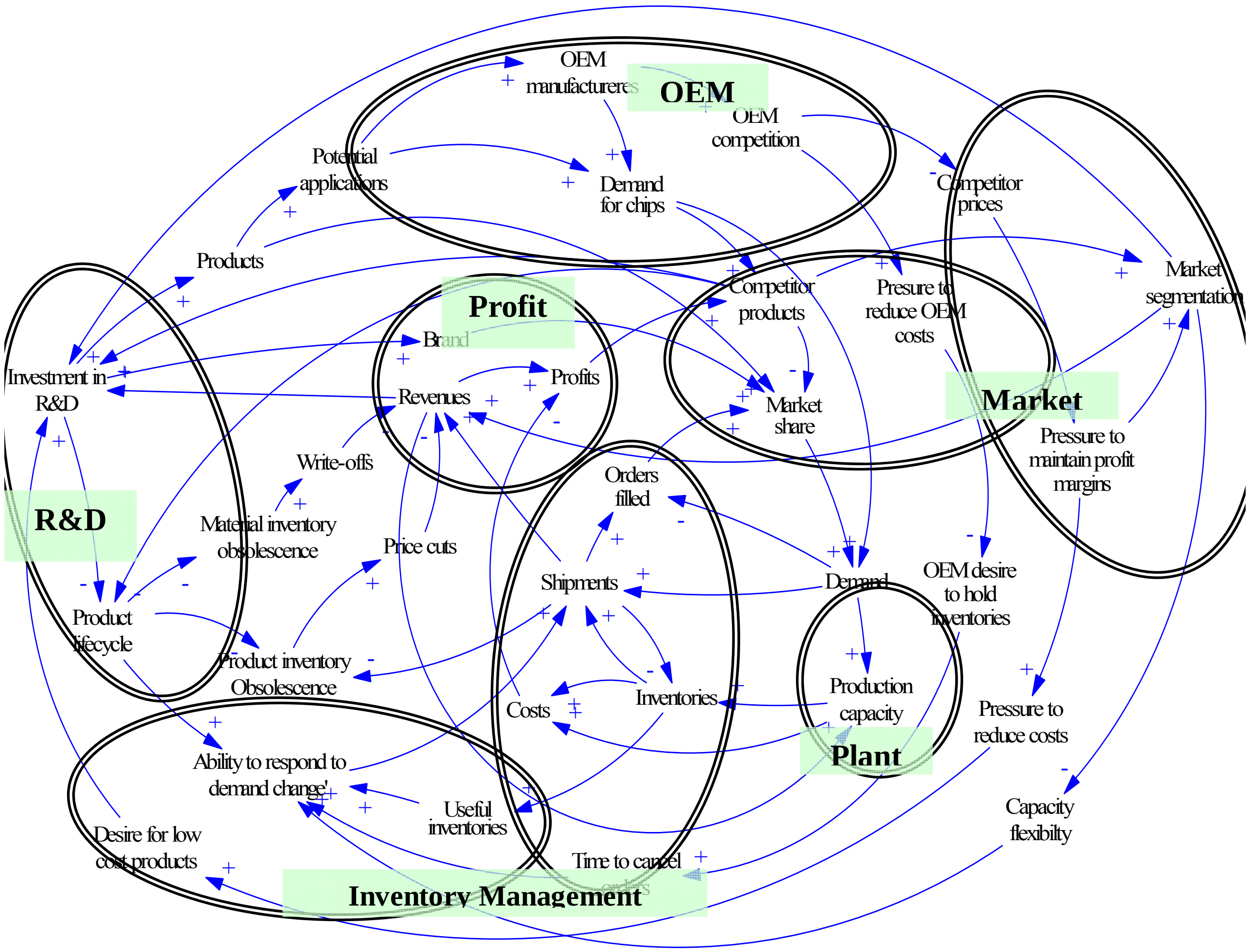


Figure 3.9 Causal loop diagram of LSMC's SC categorized in managerial action areas

3.2.3 Stocks and Flows Diagram of LSMC Supply Chain

The next step is to convert the causal loop diagrams into stocks and flows models. LSMC's supply chain model is divided into three sub-models: production, shipment, and demand forecast and capacity models. These stocks and flows models are described below. Combining all the three sub-models gives the final complete model.

3.2.3.1 Production Model

LSMC's production model was constructed based on Sterman's Production Starts model. [6] Because Sterman's production starts model is a generic model and it captures only one step production, Lertpattarapong [1] had to customize it to fit LSMC's environment. LSMC runs a push process from the pre-assembly process to the assembly process and runs a pull process from the assembly process to the packaging process where finished goods come out. LSMC's production model is shown in Figure 3.10. The Expected Channel Demand for LSMC products is a smooth function of Channel Demand for LSMC Products.

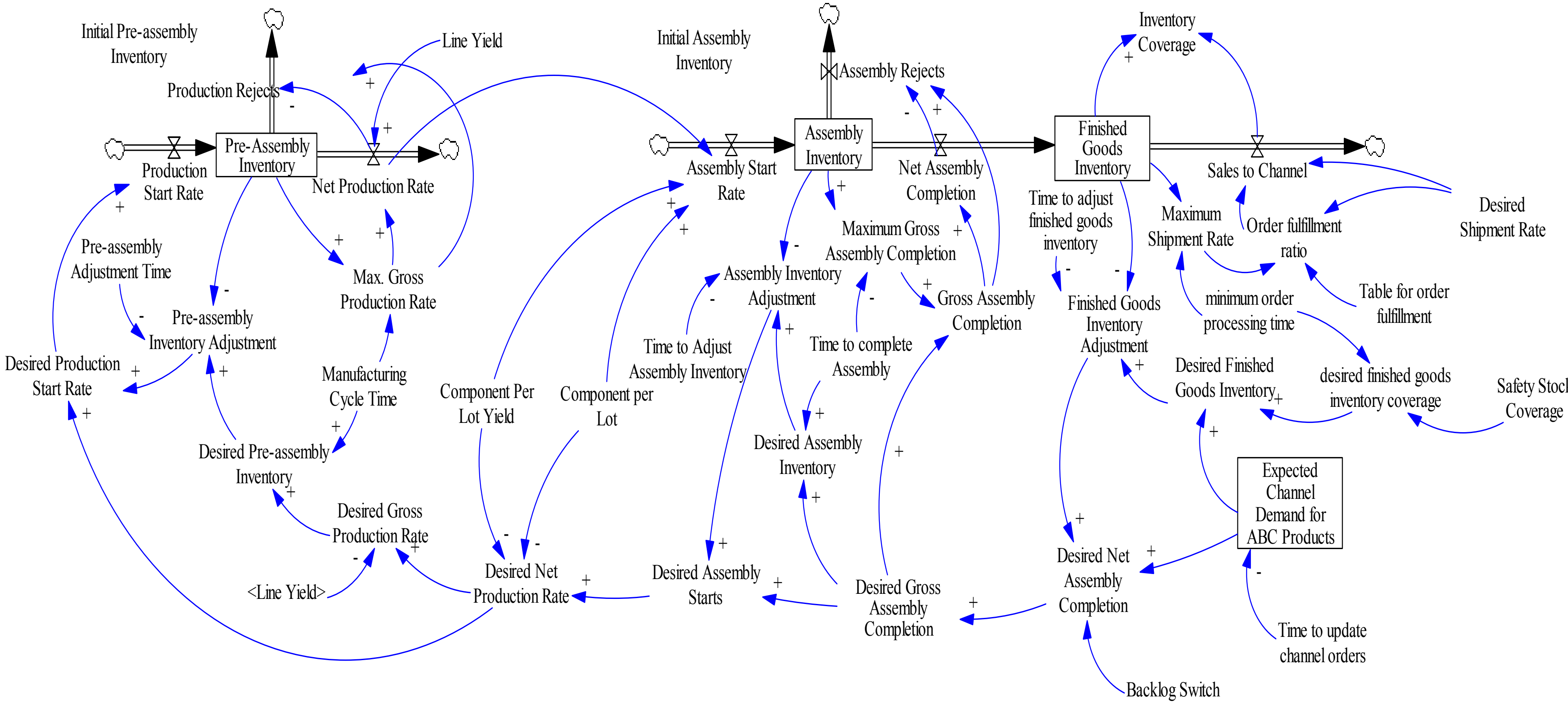


Figure 3.10 Production Model (Reference-Lertpattarapong, 2002)

3.2.3.2 Shipment Model

Shipment model, which is the second sub-model of the final model, comprises of two other sum-models, namely Inventory, Backlog and Shipping sub-model and Market Share sub-model.

1.) Inventory, Backlog and Shipping sub-model: Products re shipped to OEMs and other customers from the finished goods inventory. LSMC's Orders Filled depends on its shipment capability which is a function of the ratio of Maximum Shipment Rate to Desired Shipment Rate. When the ratio is less than one, LSMC ships products as fast as its Desired Shipment Rate. However, when the ratio is greater than one it can only ship what it has in the finished goods inventory.

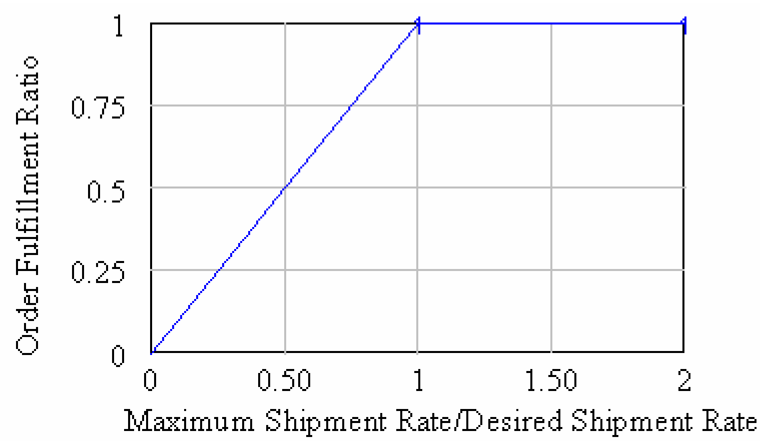


Figure 3.11 Table for Order Fulfillment

Figure 3.12 illustrates the Inventory, Backlog and Shipping sub-model.

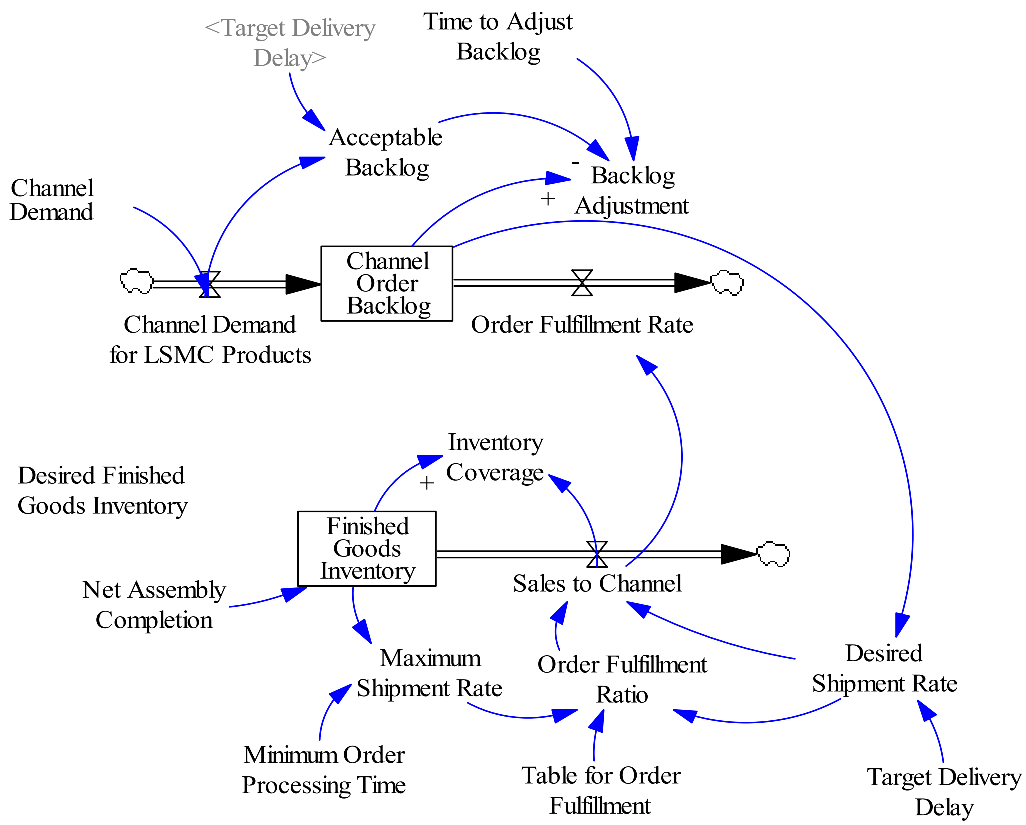


Figure 3.12 Inventory, Backlog and Shipping sub-model (Lertpattarapong, 2000)

2.) Market Share sub-model: LSMC's demand is driven by its market share and the market share is driven by LSMC's Attractiveness which is determined by how LSMC can fulfill its customer's orders. This sub-model is shown in Figure 3.13

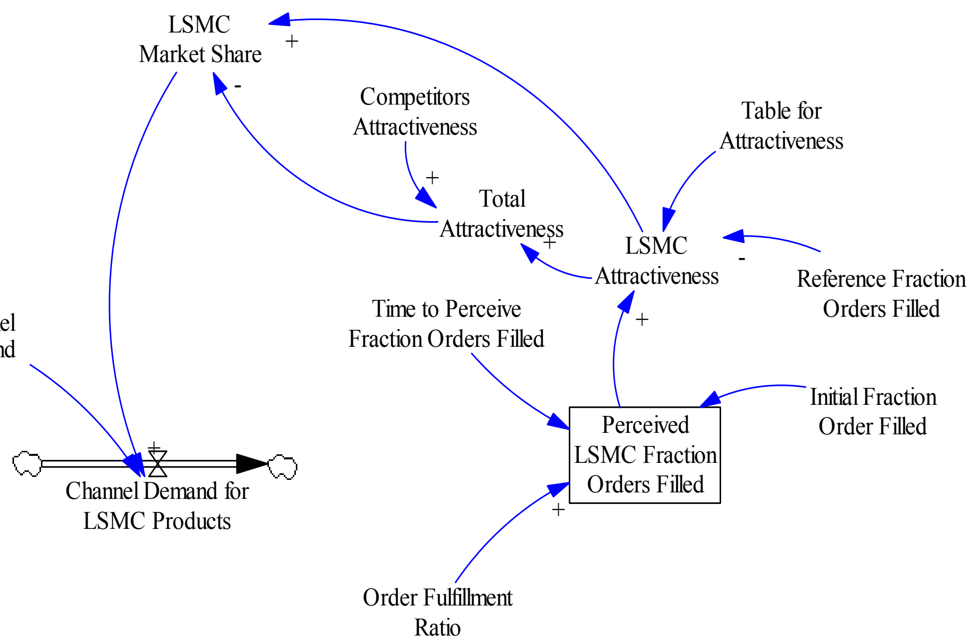


Figure 3.13 Market Share sub-model (Lertpattarapong, 2000)

Finally, combining both of Inventory, Backlog and Shipping sub-model and Market Share sub-model gives LSMC's Shipment model which is illustrated in Figure 3.14

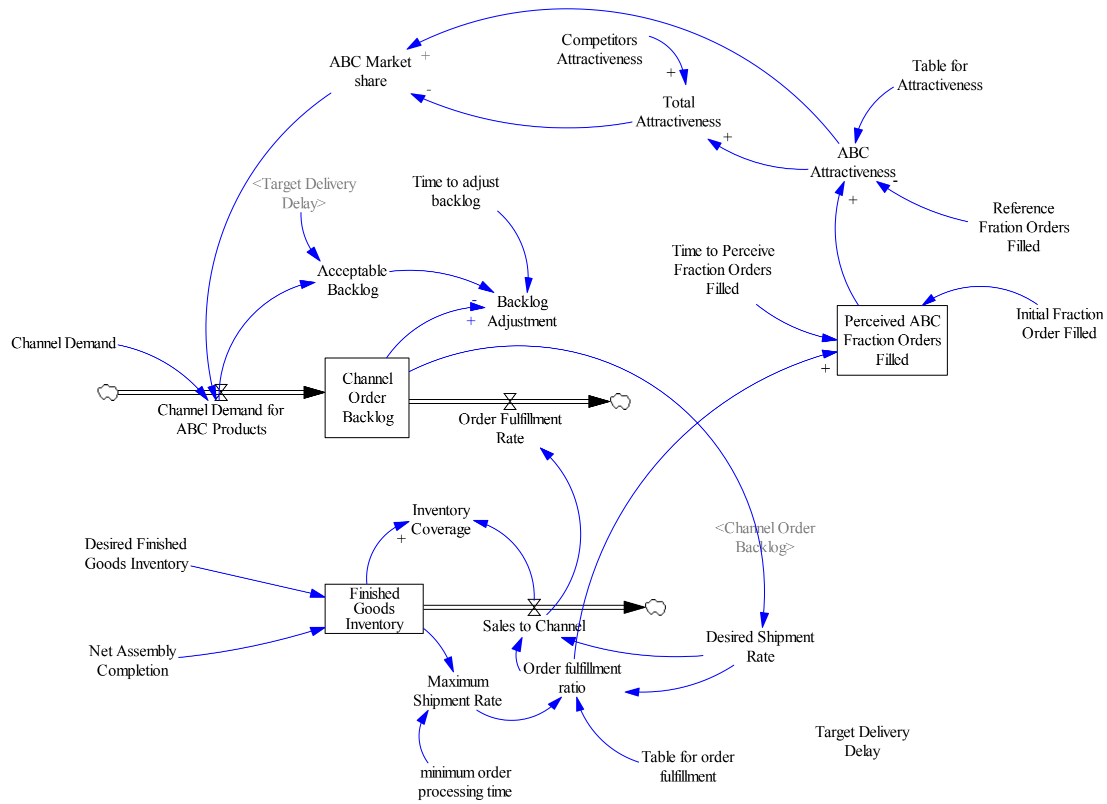


Figure 3.14 Shipment Model (Lertpattarapong, 2000)

3.2.3.3. Demand Forecast and Capacity Model

This model consists of two other sub-models, namely Demand Forecast and Capacity sub-models. Figure 3.15 illustrates the actual demand forecast and capacity model.

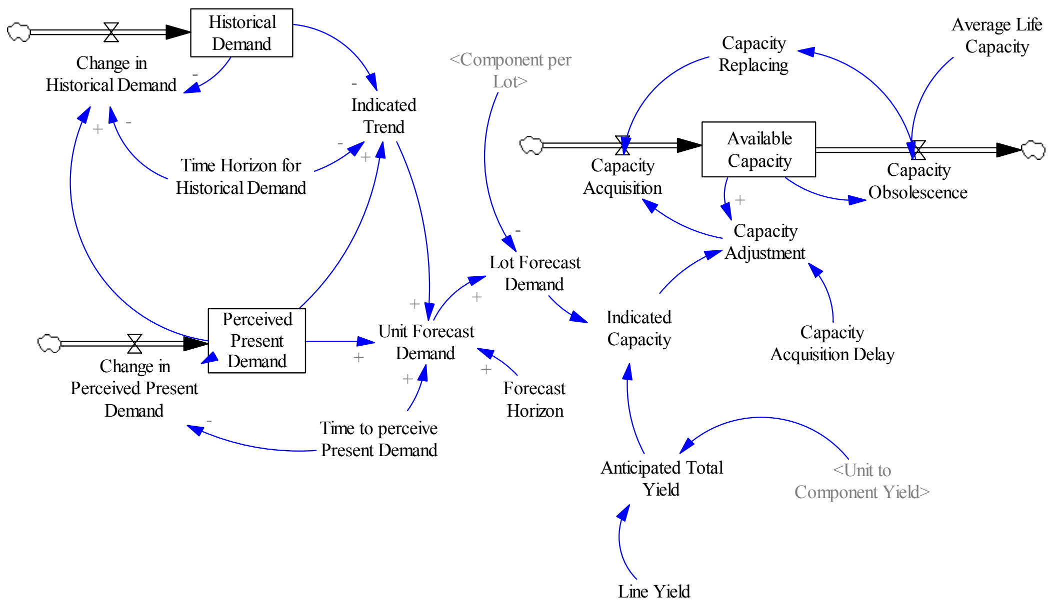


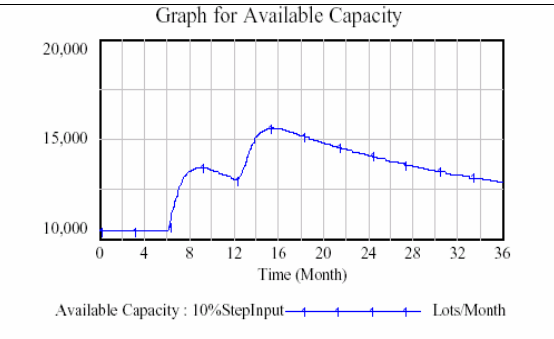
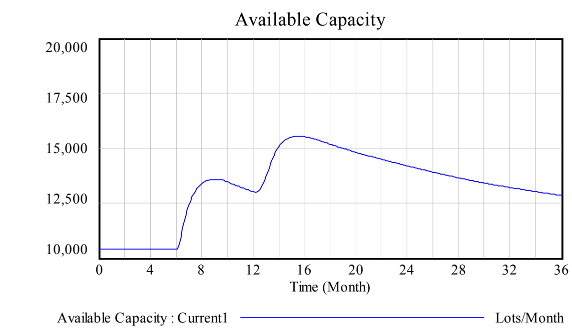
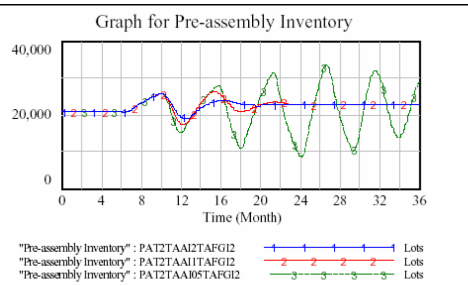
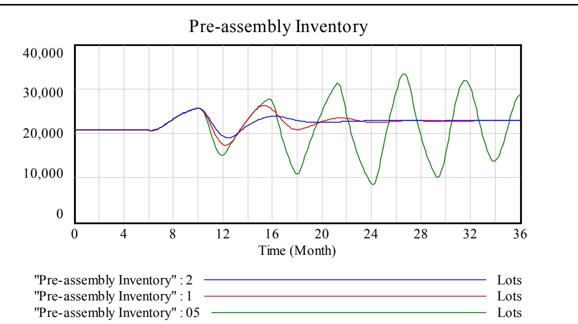
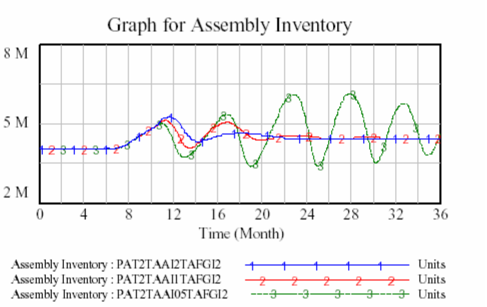
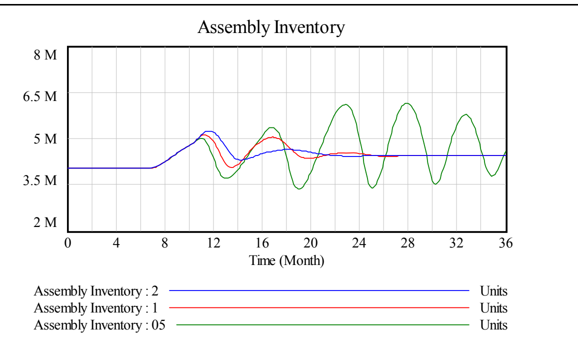
Figure 3.15 Demand Forecast and Capacity Model (Lertpattarapong, 2000)

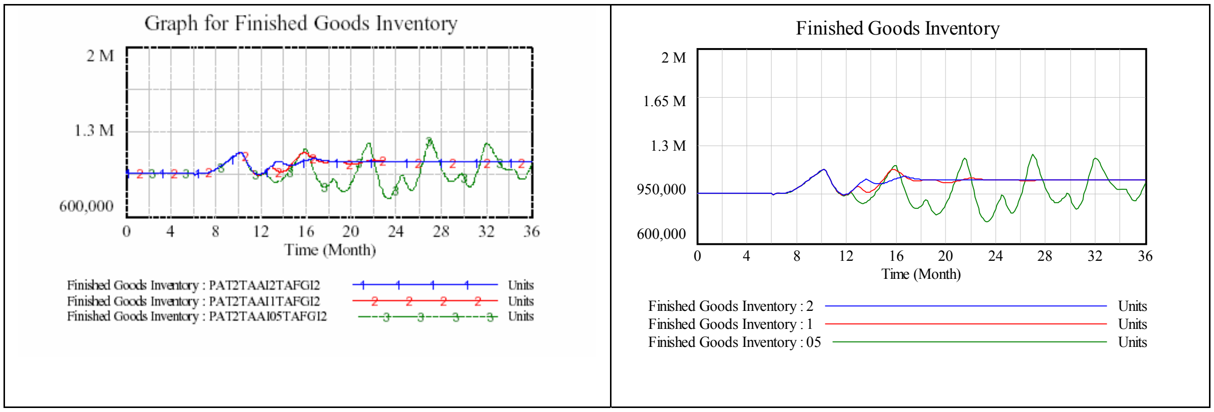
Finally, LSMC's complete supply chain model is obtained by combining Production, Shipment and Demand Forecast and Capacity sub-models.

3.3 Model Validation

The model developed by Lertpattarapong [1] for LSMC's supply chain was rebuilt in this research work using the equations he presented in his research work. This model has been formulated as a system of nonlinear differential equations. It is a very large and complicated model and there is no algebraic solution. Therefore, to check the correctness of the reproduced model, it has been simulated and the output (graphs for different variables) has been compared with that of the original model. The results and graphs compared matched with each other and hence the model is reproduced correctly. Comparison of output of some of the variables related to production and capacity is presented in table 3.1 below.

Table 3.1 Comparison of Output

<table>
<tr><th>Output from Original Model
Lertpattarapong [1]</th><th>Output from Reproduced Model
(Current)</th></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td>
</td><td>
</td></tr>
</table>



3.4 Important Observations about LSMC Supply Chain

Lertpattarapong [1] studied the dynamic behavior of the LSMC supply chain through simulations (e.g. step response simulation, ramp response simulation), and techniques like eigen value analysis and loop knockout technique. Some important observations found are as follows.

One of the observations is that varying time to adjust inventories, including Pre-assembly Adjustment Time (PAT), Time to Adjust Assembly Inventory (TAAI) and Time to Adjust Finished Goods Inventory (TAFGI), has impacts on the oscillatory behavior of the product inventories.

Another important observation is, by varying these parameters, PAT, TAAI and TAFGI, Channel Demand for LSMC products oscillates. This oscillation in Channel Demand implies that Channel Demand for LSMC products is endogenous and is caused by internal factors. Before seeing this insight, most senior managers in LSMC believed that the oscillatory demand was exogenous and the exogenous inputs caused the oscillatory behavior in product inventories.

The fluctuation in finished goods inventory (FGI) oscillates and the amplitude is large compared to demand and capacity. Also, capacity relative to desired capacity oscillates and the amplitude is growing.

Another salient observation found through ramp response simulation is that ramping production also causes oscillations in both the production inventories and Channel Demand for LSMC products.

3.5 Summary

System Dynamics modeling of LSMC's supply chain has been discussed in this chapter. Lertpattarapong had developed this model in order to determine if the oscillations taking place in product inventories (especially, finished goods inventory) and desired capacity were due to internal factors or external factors affecting the company. After the analysis, it was concluded that these oscillations were caused due to the influence of internal factors (in this case, production management policies) although they were triggered by the external factors in the OEM and Market segments of the supply chain system.

This model was rebuilt and its output was compared with the original results (graphical validation) in order to validate it for its correctness and for further use in this research work. Lertpattarapong [1] found the causes for oscillations but his research did not attempt to develop a methodology that helps mitigate/minimize these oscillations. The focus of this research work is to find out what production management policies would possibly lessen the oscillations. The development of an optimization module that would mitigate the oscillations using a genetic algorithm is presented in the following chapter.

CHAPTER 4: OPTIMIZATION MODULE DEVELOPMENT

The previous chapter dealt with a detailed discussion about LSMC's supply chain model and some important observations about the oscillations occurring in it. In this chapter, development of a single objective real-coded genetic algorithm optimization module to mitigate the oscillations in Finished Goods Inventory (FGI) of LSMC supply chain is presented.

4.1 Optimization Criteria for the problems in LSMC Supply Chain

As explained in the previous chapter, variation in time to update production inventories, such as Pre-assembly Adjustment Time (PAT), Time to Adjust Assembly Inventory (TAAI), and Time to Adjust Finished Goods Inventory (TAFGI), causes oscillations in product inventories and channel demand. In other words, production management policies were the direct cause of the oscillations observed. Another important observation is that the fluctuation in the finished goods inventory (FGI) oscillates and the amplitude is large compared to demand and capacity. This fact is utilized as the basis for developing the optimization module.

There are certain variables/parameters in LSMC's supply chain model that are independent and are in the control of LSMC { e.g. manufacturing cycle time, time to update inventories (e.g. PAT, TAAI, etc.), minimum order processing time, time to complete assembly, etc.}. Most of these variables, including those that are responsible for oscillations as discussed in the previous chapter, belong to the production and shipment models. No attempt has been made in the previous research work to find the optimal values for these variables that would considerably lessen the oscillations.

It is in this context that this research work proposes a methodology to find the optimal/near-optimal values for a set of pre-defined variables that would mitigate the oscillations. As a preliminary step, a set of five important variables has been chosen. The optimal/near-optimal values for these variables that would minimize the oscillations in finished goods inventory (FGI) will be found using a real coded genetic algorithm (RCGA). The methodology behind developing the proposed RCGA is explained in the following sections.

The five variables chosen for the purpose of minimizing the oscillations in finished goods inventory are as follows.

1. Pre-assembly Adjustment Time (PAT)
2. Time to Adjust Assembly Inventory (TAAI)
3. Time to Adjust Finished Goods Inventory (TAFGI)
4. Manufacturing Cycle Time (MCTime)
5. Minimum Order Processing Time (MOPTime)

4.2 Development of Real-Coded Genetic Algorithm

The working principle of a conventional genetic algorithm has been explained in Section 2.3.1 of Literature Review. A real-coded genetic algorithm (RCGA) also works on the same principles but the only difference lies in the genetic operators used.

The RCGA presented in this thesis has been developed and implemented in C++ language. The basic structure and genetic operators of this algorithm have been adopted from the real-coded algorithm developed by Dr. Deb Kalyanmoy [47].

The basic objective behind developing this algorithm is to find the optimal/near-optimal values for the set of five variables mentioned in the previous section that would minimize the oscillations in the finished goods inventory (FGI) of LSMC supply chain. The graph of FGI for the original LSMC model looks as is shown in Figure 4.1. The oscillations in the curve can be very clearly seen.

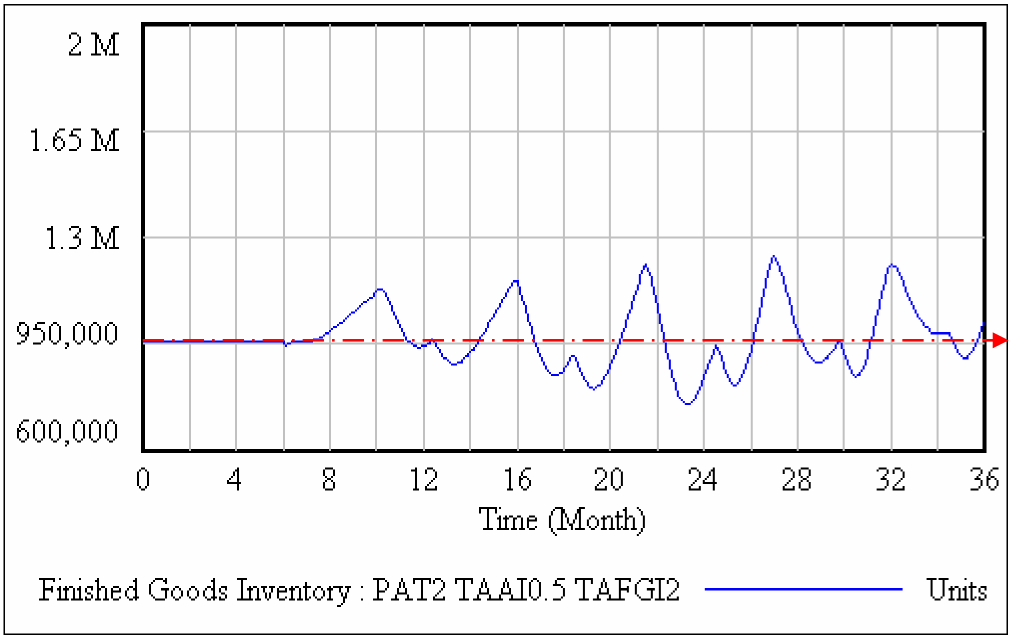


Figure 4.1 Finished Goods Inventory of LSMC Supply Chain

The criterion used to minimize these oscillations in FGI is to minimize the area under this curve. An imaginary axis (shown in red in Figure 4.1) is imagined at the initial condition and the absolute value of the area under the curve about this imaginary axis is minimized. When the area under the curve is zero, the curve is just a straight line meaning that FGI remains constant over time. The RCGA developed takes as input, the lower and upper bounds of the five variables mentioned above, and attempts to find the optimal/near-optimal values (for these five variables) that would give rise to a FGI curve with the minimum area possible.

The following sub-sections present an explanation of the working steps of the developed algorithm, a brief description of the C++ program and how it is integrated with the LSMC's system dynamics supply chain model in Vensim, and how to run the actual program to obtain the result.

4.2.1. Working Procedure of the RCGA

The following is a description of the sequential steps involved in the working of the proposed algorithm. The flowchart presented in Figure 4.2 clearly depicts all these steps as they are implemented by the C++ program.

1. *Reading the Input File:* The very first step in the algorithm is to read the input from the input file. The input file is a simple notepad document containing the parameters for controlling the algorithm (e.g. number of generations, population size, etc.), and the lower and upper limits of the five variables (MCTime, MOPTime, TAAI, PAT, TAFGI). The input file is controlled by the user. (More details about the input file are presented in Section 4.3.1).
2. *Initializing the population:* The next step in the algorithm is to randomly create an initial population. This population contains as many individuals as the Population Size mentioned in the input file.

   Each individual is a set of five values, one for each of the five variables (MCTime, MOPTime, TAAI, PAT, and TAFGI), generated within the specified lower and upper bounds of that variable.
3. *Fitness evaluation and new population creation:* The following sub-steps are iteratively performed on each generation of the population until a specified number of runs of the algorithm are performed.

(A.) The fitness of each individual in the population is evaluated. Fitness of an individual in this case is nothing but the absolute value of the area of the FGI curve generated by simulating the supply chain model with the values of the variables corresponding to that individual. (For calculating the area of the FGI curve, a new variable (AreaFGI) has been added to the original model that integrates the FGI curve between the specified time limits to compute the area under the curve).

Actually, the C++ program, after creating each generation of the population, sends each individual (set of five values) in the population to the Vensim simulation model by means of VensimDLL (Vensim Dynamic Link Library) and simulates the model with these values, and retrieves the area of the FGI curve. Then, the area of the curve corresponding to that individual is assigned as its fitness.

(B.) At the end of each generation, the fitness of all individuals in that generation is compared and the individual that is identified to be having the best fitness (i.e. the least area for FGI curve) is designated as the result of the algorithm for that generation.

(C.) *Creating new population:* A new population of individuals is created by applying the following three genetic operators. The genetic operations are applied to the individuals in the population chosen with a probability based on their fitness.

(i.) *Selection or Reproduction:* The primary objective of reproduction operator is to make duplicates of good solutions and eliminate bad solutions in a population, while keeping the population size constant. The proposed algorithm uses Stochastic Remainder Roulette-Wheel (SRRW) selection operator to create a new population. This operator is explained in the next sub-section.

(ii.) *Crossover:* A crossover operator called Simulated Binary Crossover (SBX) with a pre-defined probability (P_c=0.5) is applied next to the variables of the mating pool in order to produce new solutions. SBX operator is explained in the next sub-section.

(iii.) *Mutation:* The need for mutation is to keep diversity in the population. After the use reproduction and crossover, polynomial mutation operator, explained in the next sub-section, is applied in order to induce some diversity in the population.

After a new population has been created using the genetic operators, the fitness evaluation of the population is performed (as mentioned in sub-step A) followed by identifying the best individual for that generation of the algorithm (sub-step B).

4. After every single run of the algorithm, the individual that is identified to be having the best fitness (among all the generations in this run) is designated as the result of the algorithm for that run. After all the specified number of runs of the algorithm are performed, the individual having the best fitness among all the runs is designated as final result of the algorithm, which represents a solution (or an approximate solution) to the problem.

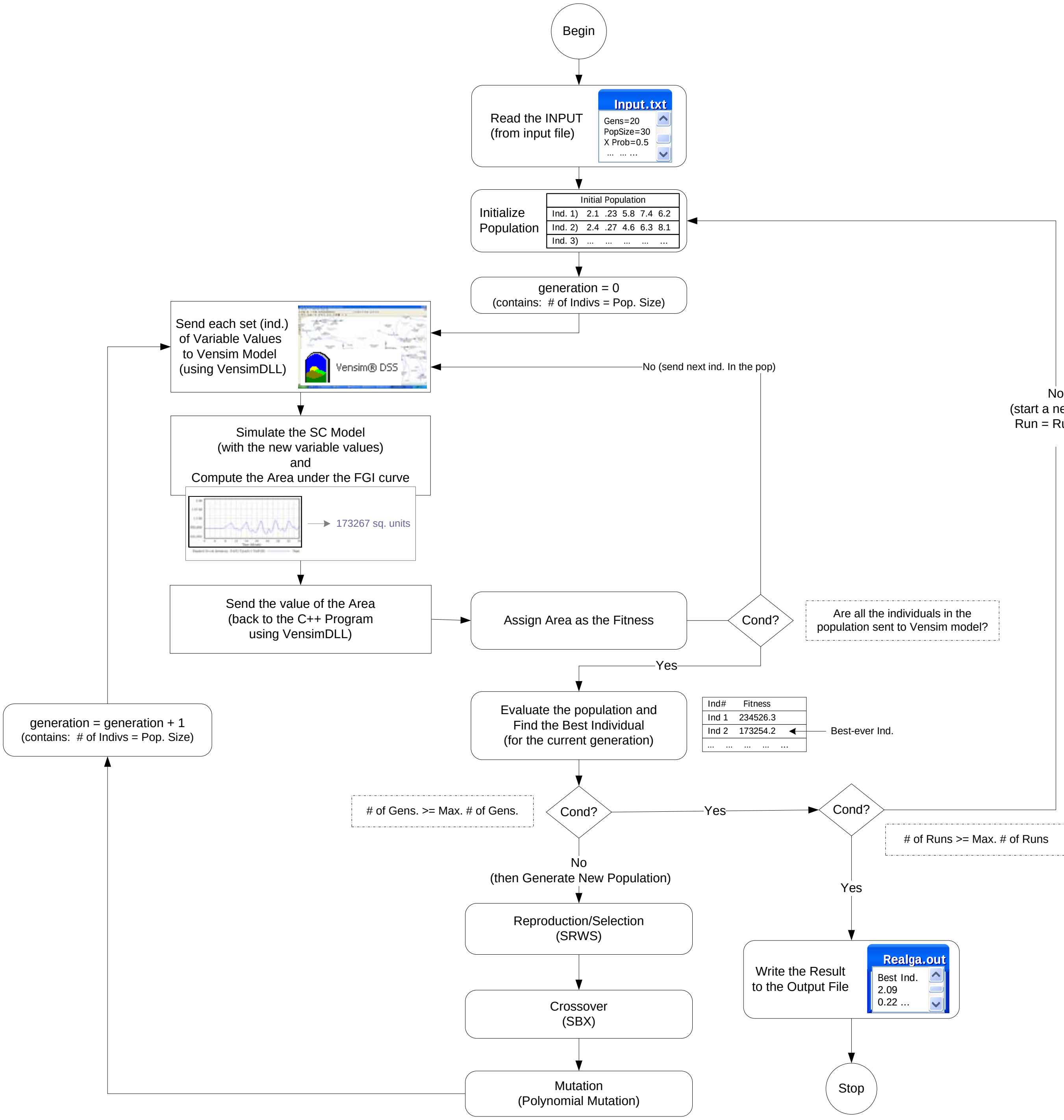


Figure 4.2 Working Steps of the RCGA in the C++ Program

4.2.2. Genetic Operators Used in the RCGA

This section describes the genetic operators used in the RCGA for selection, crossover and mutation operations.

1. Stochastic Roulette-Wheel Section (SRWS): In order to explain SRWS, roulette-wheel selection (RWS) operator is first explained. In RWS, solutions are assigned copies, the number of which is proportional to their fitness. If the average fitness of all the population members is f_{avg}, a solution with a fitness f_i gets an expected number of copies equal to f_i/f_{avg}. The implementation of this selection operator can be thought of as a roulette-wheel mechanism, where the wheel is divided into N (population size) divisions, where the size of each is marked in proportion to the fitness of each population member. Thereafter, the wheel is spun N times, each time choosing the solution indicated by the pointer of the roulette-wheel. This operation can be easily simulated on a computer. Using the fitness value of all the solutions, the probability of selecting the i^{th} solution is $p_i = f_i / \sum_{j=1}^{N} f_j$. Thereafter, the cumulative probability ($P_i = \sum_{j=1}^{i} p_j$) of each solution is calculated by adding the individual probabilities from the top of the list of solutions. Thus, the bottom-most in the population has a cumulative probability (P_N) equal to 1. The roulette-wheel concept can be simulated by realizing that the i^{th} solution in the population represents the cumulative probability values in the range $[P_{i-1}, P_i]$. In order to choose N solutions, N random numbers between zero and one are generated. Thus, a solution that represents the chosen random number in the cumulative probability range (calculated from the fitness values) for the solution is copied to mating pool.

The above implementation of the RWS is noisy in the sense of introducing a large variance in its realizations. This variance may be reduced by using a somewhat deterministic version of the RWS operator, called Stochastic Roulette-Wheel Selection (SRWS) operator. In SRWS, the probabilities p_i are multiplied by the population size and the expected number of copies is calculated for all solutions. Thereafter, each solution is first assigned a number of copies equal to the integer part of the expected number. Thereafter, the usual roulette-wheel selection operator is applied to the fractional part of the expected number of all solutions to assign further copies. Since a part of the assignment process is deterministic, this operator is less noisy.

2. Simulated Binary Crossover (SBX): Deb [43] developed SBX operator, which works on two parent solutions and creates two offspring. SBX operator simulates the working principle of the single-point crossover on binary strings. The procedure of computing the offspring $X_i^{(1, t+1)}$ and $X_i^{(2, t+1)}$ from parent solutions $X_i^{(1, t)}$ and $X_i^{(2, t)}$ is described as follows.

   A spread factor (β_i) is defined as the ratio of the absolute difference in offspring values to the parents: $\beta_i = \left| \dfrac{X_i^{(2, t+1)} - X_i^{(1, t+1)}}{X_i^{(2, t)} - X_i^{(1, t)}} \right|$

   The following is a step-by-step procedure to create the offspring from the parents.

   Step 1: Choose a random number $U_i \in [0, 1)$.

   Step 2: From a specified probability distribution function, the ordinate $\beta_{q\,i}$ is found so that the area under the probability curve from 0 to $\beta_{q\,i}$ is equal to the chosen random number U_i. The probability distribution used to create the offspring is as follows [43]:

$P(\beta_i) = 0.5*(\eta + 1)*(\beta_i)^{\eta}$, if $\beta_i \leq 1$

$= 0.5*(\eta + 1)*(1/\beta_i^{\eta+2})$, otherwise; where η is a non-negative real number.

After equating the area under the above probability curve to U_i, the value of $\beta_{q\,i}$ is given as follows.

$\beta_{q\,i} = (2U_i)^{1/\eta+1}$, if $U_i \leq 0.5$

$= [2(1-U_i)]^{-1/\eta+1}$, otherwise

Step 3: After computing β_{qi}, the offspring are computed using the following equations.

$X_i^{(1,\,t+1)} = 0.5\ \{(1 + \beta_{q\,i})\, X_i^{(1,\,t)} + (1- \beta_{q\,i})\, X_i^{(2,\,t)}\}$,

$X_i^{(2,\,t+1)} = 0.5\ \{(1 - \beta_{q\,i})\, X_i^{(1,\,t)} + (1 + \beta_{q\,i})\, X_i^{(2,\,t)}\}$.

A value of η=2 has been used in the actual algorithm implementation.

3. Polynomial Mutation: A mutated solution $Y_i^{(1,\ t+1)}$ using polynomial mutation is obtained as follows.

$Y_i^{(1,\,t+1)} = X_i^{(1,\,t+1)} + \{Xi^{(U)} - Xi^{(L)}\}\delta_i$, where δ_i is calculated from the polynomial probability distribution $P(\delta) = 0.5\,(\eta+1)\,(1-|\delta|)^{\eta}$ as follows:

$\delta_i = (2.r_i)^{(1/\eta+1)} - 1$, if $r_i < 0.5$,

$= 1 - [2(1- r_i)^{(1/\eta+1)}$, if $r_i \geq 0.5$. {Random number, $r_i \in [0, 1)$}

A value of η=20 has been used in the actual algorithm implementation.

4.3 The C++ Program for RCGA and its Integration with Vensim System Dynamics Model

A computer program in C++ language has been developed to implement the algorithm. The working steps of this program are presented in Figure 4.2. The basic structure and genetic operators for this program have been adopted from the code developed by Dr. Deb [47]. The following is a screen shot of the actual program developed in Microsoft Visual Studio.

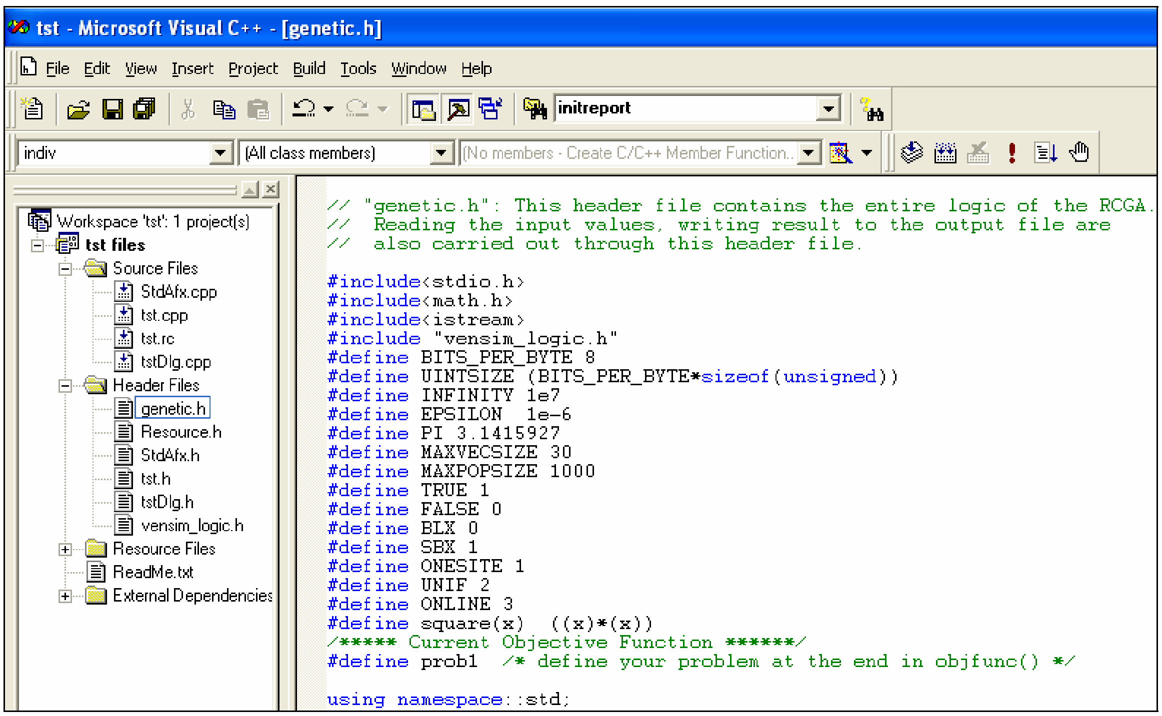


Figure 4.3 A screen shot of the C++ program

This program is comprised of a couple of Source files (e.g. tst.cpp) and Header files (e.g. genetic.h). The main logic of the algorithm is contained in *tst.cpp*, *genetic.h* and *vensim_logic.h*. The rest of the files are just the supporting files.

tst.cpp: This is the main application source file. In order to run the algorithm, this program must be built and executed. This program initiates the algorithm by calling the functions in the header files *genetic.h* and *vensim_logic.h*; First the sub-routine 'load_model()' in the header file *vensim_logic.h* is called. This sub-routine loads the specified vensim model (LSMC supply chain model), ready to be simulated with the values generated by the genetic algorithm. Upon successful loading of the specified model, the sub-routine 'this_main()' in the header file *genetic.h* is called. This sub-routine starts the genetic algorithm.

vensim_logic.h: This header file contains the logic to load the vensim simulation model and passing the values of the five variables (MCTime, MOPTime, TAAT, PAT, and TAFGI) generated by the genetic algorithm, simulate the model with these values and finally retrieve the value of the absolute value of the area under the finished goods inventory curve. The retrieved value of the area is sent to *genetic.h* for fitness evaluation.

The process of sending and retrieving the values to and from the vensim model is carried out by means of Vensim Dynamic Link Library (DLL). The Vensim DLL is a separate program (that comes with the software itself) that can be called from other applications such as Visual Basic, Visual C++, Excel, Delphi etc. in order to be able to use the models developed in Vensim. It is very important to include the *vendll.h* header file (that comes with software itself) in the C++ program whenever an application is making use of Vensim DLL [48].

genetic.h: This header file contains the actual logic for the genetic algorithm. The parameters for the GA and the limits for the five variables are given through as input file called 'input.txt' (explained in next section). The logic for reading the values from this input file is present in *genetic.h*. After reading the values, the genetic algorithm is run (all the steps mentioned in Section 4.2.1) over a specified number of runs and the output is written to an output file called ‘realga.out’.

The input data to be presented to the algorithm and the format of the output result are discussed in the next section.

4.3.1. Running the Algorithm

The section presents the format of the data to be presented to the input file, how to run the C++ program and how to see the output.

Input File (input.txt): Input file is a simple notepad file (.txt) through which the program reads certain values. Before running the algorithm the parameters for the genetic algorithm (e.g. number of runs, number of generations, etc.) and the limits (lower and upper limits) for the five variables are to be specified through this input file. An example input file is shown in the Figure 4.3.

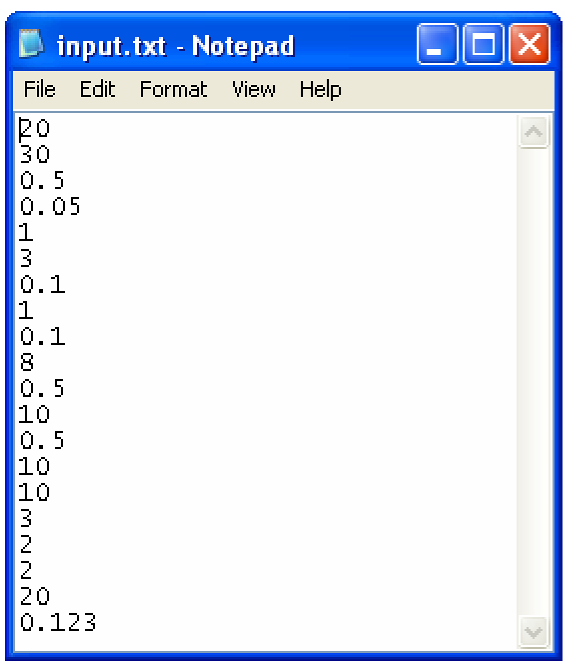
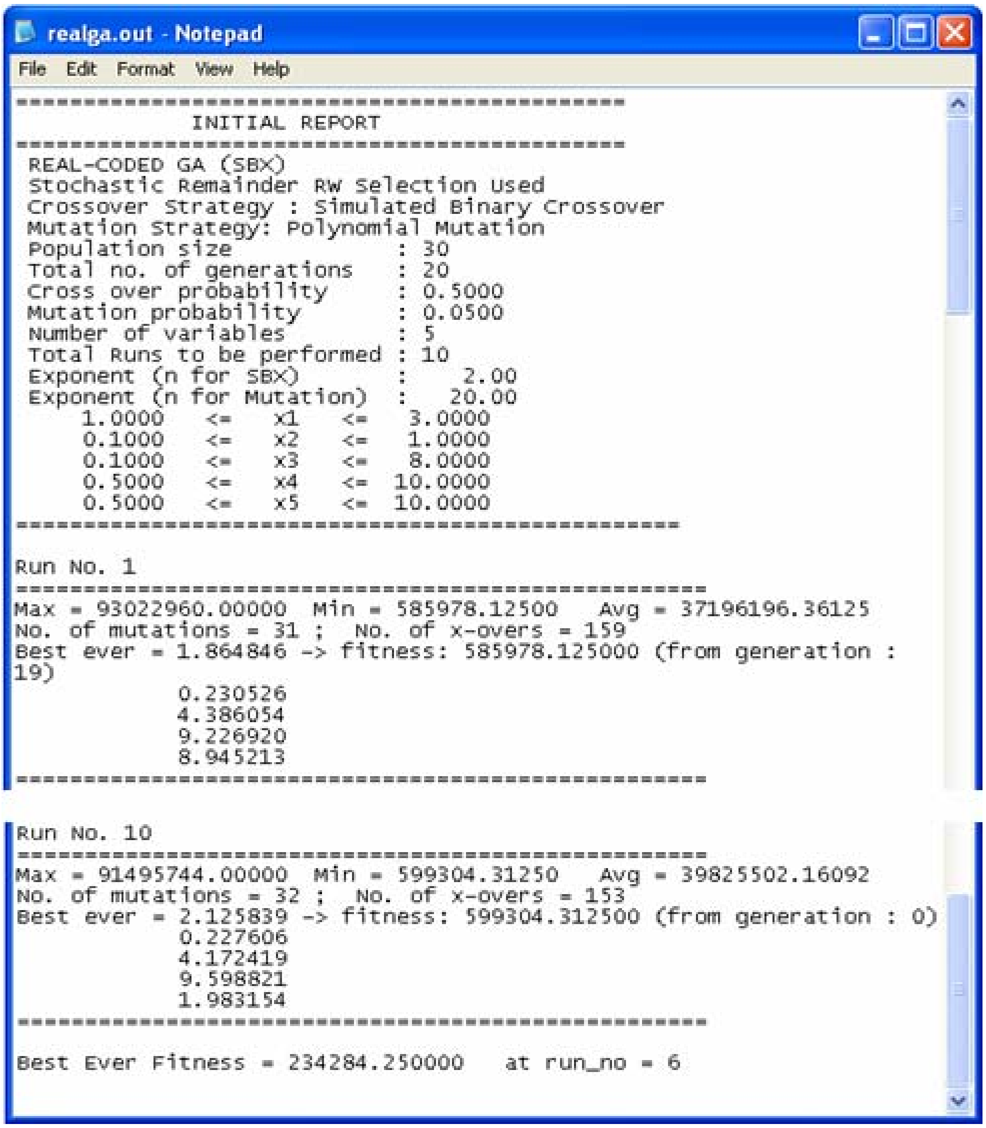


Figure 4.4 Example Input File

It is very important to input these values only in the order mentioned below. This is the exact order in which the program reads the values.

1. Number of Generations
2. Population Size
3. Probability of Crossover
4. Probability of Mutation
5. Lower Limit for MCTime (Manufacturing Cycle Time)
6. Upper Limit for MCTime
7. Lower Limit for MOPTime (Minimum Order Processing Time)
8. Upper Limit for MOPTime
9. Lower Limit for TAAI (Time to Adjust Assembly Inventory)
10. Upper Limit for TAAI
11. Lower Limit for PAT (Pre Assembly Adjustment Time)
12. Upper Limit for PAT
13. Lower Limit for TAFGI (Time to Adjust Finished Goods Inventory)
14. Upper Limit for TAFGI
15. Number of Runs
16. Selection Strategy (=3 by default, for Stochastic Roulette Wheel Selection)
17. Crossover Strategy (=2 by default, for Simulated Binary Crossover)
18. Exponent (η) for Crossover (A value of $\eta = 2$ is used)
19. Exponent (η) for Mutation (A value of $\eta = 20$ is used)
20. Random seed (A value between 0 and 1, required to start the initialization process)

Running the Program: The next step after inputting the values is to run the program. As mentioned earlier, *tst.cpp* is the main application source file. For running the program the executable file *tst.exe* has to built and run. Once the program is successfully run, the output is written into the file 'realga.out'.



```
realga.out - Notepad
File Edit Format View Help
==============================================
               INITIAL REPORT
==============================================
 REAL-CODED GA (SBX)
 Stochastic Remainder RW Selection Used
 Crossover Strategy : Simulated Binary Crossover
 Mutation Strategy: Polynomial Mutation
 Population size            : 30
 Total no. of generations   : 20
 Cross over probability     : 0.5000
 Mutation probability       : 0.0500
 Number of variables        : 5
 Total Runs to be performed : 10
 Exponent (n for SBX)       :    2.00
 Exponent (n for Mutation)  :   20.00
      1.0000   <=    x1   <=    3.0000
      0.1000   <=    x2   <=    1.0000
      0.1000   <=    x3   <=    8.0000
      0.5000   <=    x4   <=   10.0000
      0.5000   <=    x5   <=   10.0000
==================================================

Run No. 1
====================================================
Max = 93022960.00000  Min = 585978.12500   Avg = 37196196.36125
No. of mutations = 31 ;  No. of x-overs = 159
Best ever = 1.864846 -> fitness: 585978.125000 (from generation :
19)
             0.230526
             4.386054
             9.226920
             8.945213
====================================================

Run No. 10
====================================================
Max = 91495744.00000  Min = 599304.31250   Avg = 39825502.16092
No. of mutations = 32 ;  No. of x-overs = 153
Best ever = 2.125839 -> fitness: 599304.312500 (from generation : 0)
             0.227606
             4.172419
             9.598821
             1.983154
====================================================

Best Ever Fitness = 234284.250000   at run_no = 6
```

Figure 4.5 Example Output File

Output File (realga.out): The output file 'realga.out' generated after running the program looks as is shown in the Figure 4.4. The statistics (maximum, minimum and best ever fitness) for every run of the algorithm and the values of the five variables (MCTime, MOPTime, TAAI, PAT and TAFGI in that order) which produced the best fitness (minimum area) for that run are reported in the output file. Also, the run which has the best fitness (Best-ever fitness) among all the runs is presented at the end of the report. The variable values corresponding to the run, which has the best-ever fitness represent a solution to the problem.

After obtaining the variable values of the best-ever fitness run, these values are substituted in the original LSMC model and is run in order to compare the output of the original LSMC supply chain model with that obtained using the values from the RCGA.

4.4 Results and Analysis for LSMC Model

The genetic algorithm for LSMC model was run using the input values mentioned below.

1. Number of Generations = 20
2. Population Size = 30
3. Probability of Crossover = 0.5
4. Probability of Mutation = 0.05
5. Lower and Upper Limits for MCTime = (1, 3)
6. Lower and Upper Limits for MOPTime = (0.1, 1)
7. Lower and Upper Limits for TAAI = (0.1, 8)
8. Lower and Upper Limits for PAT = (0.5, 10)
9. Lower and Upper Limits for TAFGI = (0.5, 10)
10. Number of Runs = 30
11. Selection Strategy (=3 by default, for Stochastic Roulette Wheel Selection)
12. Crossover Strategy (=2 by default, for Simulated Binary Crossover)

13. Exponent (η) for Crossover (A value of $\eta = 2$ is used)

14. Exponent (η) for Mutation (A value of $\eta = 20$ is used)

15. Random seed = 0.123

The values of the five variables obtained for the RCGA using the above values as the input are as follows.

1. Manufacturing Cycle Time = 2.090522 months
2. Minimum Order Processing Time = 0.229353 months
3. Time to Adjust Assembly Inventory = 6.240052 weeks
4. Pre Assembly Adjustment Time = 2.612051 weeks
5. Time to Adjust Finished Goods Inventory = 8.229517 weeks

A comparison of the values of the five variables obtained from the genetic algorithm with that of the original LSMC model values is presented in Table 4.1.

Table 4.1 Comparison of new variable values with the original values

	New Values from GA	Original Values from LSMC Model
Manufacturing Cycle Time	2.090522 months	2 months
Minimum Order Processing Time	0.229353 months	0.25 months
Time to Adjust Assembly Inventory	6.240052 weeks	0.5 weeks
Pre Assembly Adjustment Time	2.612051 weeks	2 weeks
Time to Adjust Finished Goods Inventory	8.229517 weeks	2 weeks

The finished goods inventory curve obtained after simulating the LSMC model with these new variable values is shown in the Figure 4.5. The curve shown in blue is the original curve. The new curve (shown in red) obtained using the proposed algorithm is relatively better than the original one in terms of the oscillations occurring in the finished goods inventory. Hence, the proposed methodology, though in its preliminary stage, is quite capable of minimizing the oscillations in finished goods inventory.

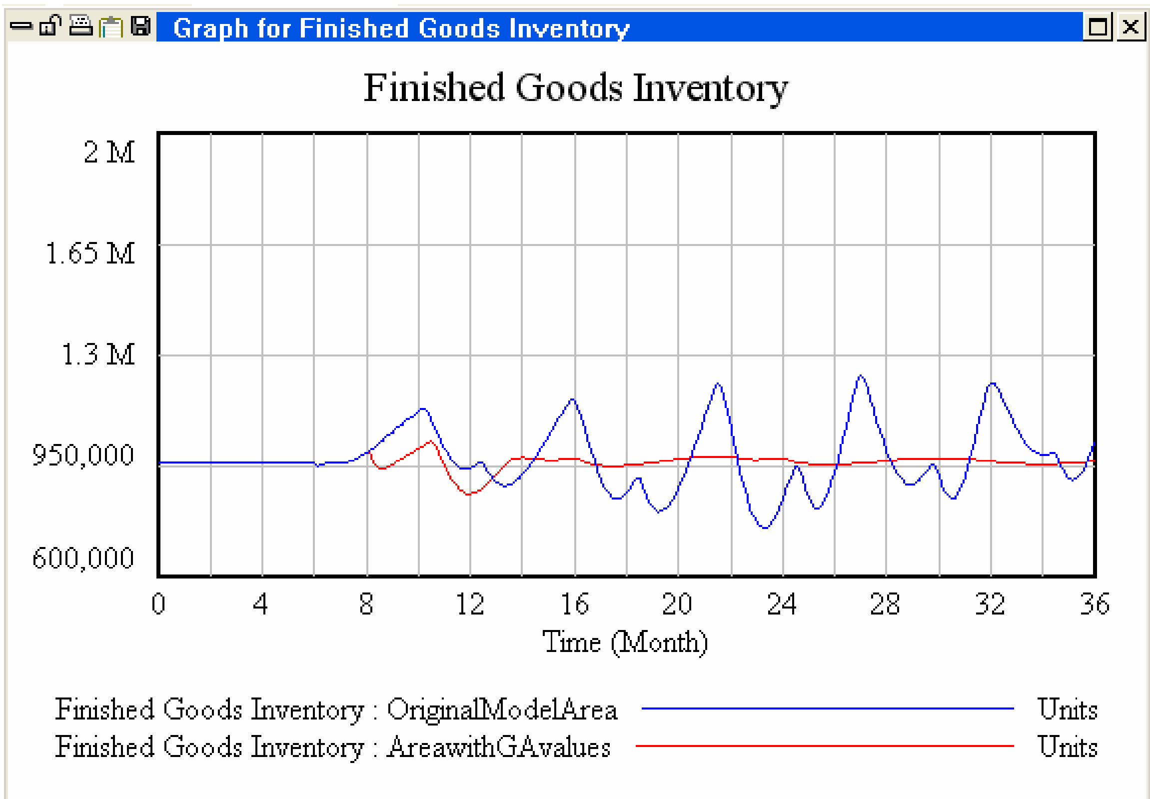


Figure 4.6 Finished Goods Inventory curve with reduced oscillations

An interesting point about the new values is that the Manufacturing Cycle Time does not vary much which is a good sign. Also, from the comparison of the values (Table 4.1), it is evident that LSMC has to reduce its Minimum Order Processing time and needs more time to adjust inventories (TAAI, PAT and TAFGI) in order to be able to have less oscillation in the finished goods inventory.

CHAPTER 5: CONCLUSION

5.1 Conclusion

The objective of this thesis is to develop a unique methodology to minimize the oscillations occurring in a supply chain by combining the techniques of System Dynamics and Genetic Algorithms. The proposed methodology utilizes the modeling flexibilities of system dynamics to model complex systems and the capability of genetic algorithms to search complex and non-linear search spaces. Utilizing the traditional optimization techniques to accomplish the same task would be much more difficult and tedious.

This methodology, though is in its preliminary stage, definitely is promising and has a high potential to be developed into an effective SCM solution with the use of advanced genetic operators and other latest developments in genetic algorithms.

5.2 Contribution of the Thesis

System Dynamics has been well recognized and widely accepted and used by many researchers as an efficient technique for modeling complex systems. Many applications have been found in the literature that involve extensive use of system dynamics to model complex supply chains in order to understand their dynamics and devise effective policies for improved performance. On the other hand, Genetic Algorithms are well known for their optimization capabilities in complex search spaces. GAs have been explored to a great extent for solving a wide variety of complex optimization problems not only in SCM but also in many other fields.

Inspite of their popularity and capability, no application was found in the literature that combined both of these techniques, exploiting their combined power, towards improving the performance of a supply chain. The methodology proposed in this research work contributes to fill this gap. This idea also provides a good field to delve in for further research and work to come.

5.3 Scope of the Thesis

This thesis utilizes expertise mainly from the fields of system dynamics and genetic algorithms. The concepts of feedback loops and dynamic modeling have been adapted from system dynamics. The methodology described to minimize the oscillations in a supply chain by stochastically searching for near-optimal solution has been developed using the concepts of genetic algorithms. The methodology presented in this thesis is a novel approach to solve a supply chain problem. There is immense potential for the extension of this work, in order to make a more robust approach. The following section presents a few potential ideas for future work.

5.4 Ideas for Potential Future Work

A couple of vital improvements, including but not limited to the ones mentioned below, can be made to the genetic algorithm employed to minimize the oscillations.

- Employing new genetic operators and search concepts: Making use of latest advances in genetic operators and implementing the latest search concepts like non-dominated sorting [43] would make the proposed approach more valuable.
- Number of variables: The proposed methodology uses only five variables which are considered very important and vital in minimizing the oscillations of finished

goods inventory. However, considering each and every variable that would affect the problem would be crucial in developing a more robust algorithm.

- Multi-objective Genetic Algorithm: The proposed methodology takes into consideration only the objective of minimizing the oscillations in finished goods inventory, whereas developing a multi-objective optimization algorithm would be very interesting to the context of a real-world business setting. This is very important because we would like to optimize not only Finished Goods Inventory but also Assembly Inventory and other supply chain parameters.
- Handling Constraints: Adding the capability of handling the user specified constraints to the present algorithm would be a good improvement.
- Genetic algorithms and eigen-value analysis: The concept of eigen-value analysis and eigen-value elasticity are used to analyze the behavior of linear dynamic systems. These concepts have also been used to analyze and understand complex supply chains. Combining eigen-value analysis and genetic algorithms to analyze supply chain oscillations could provide interesting insights.

REFERENCES

1. Lertpattarapong C., *"Applying system dynamics approach to the supply chain management problem"*, Masters Thesis, Sloan School of Management, M.I.T. 2002.
2. Ankit S. Shah, *"Detecting internal and external changes in a supply chain and predicting its behavior using neural networks"*, Masters Thesis, UCF, 2003.
3. Ashish Dhar, *"Analysis of Oscillations in supply chains"*, Masters Thesis, University of Central Florida, 2003.
4. Rabelo, L., Magdy Helal and Lertpattarapong C., "Pattern recognition in supply chain management".
5. Rabelo, L., Magdy Helal and Lertpattarapong C., *"A system dynamics based procedure for predicting structural changes in supply chains behavior"*, Proceedings of the 2004 Winter Simulation Conference, 2004.
6. Sterman, J.D., *"Business Dynamics: Systems thinking and modeling for a complex world"*, McGraw-Hill, Boston, 2000.
7. Tracy, M., Fite, W. R., and Sutton, J. M., *"An explanatory model and measurement instrument: a guide to supply chain management research and applications"*, American Journal of Business, Fall 2004, 19, 2.
8. Mohideen, F., "Ascertaining the growth of a company: a system dynamics approach", Masters Thesis, University of Central Florida, Fall 2004.
9. J Sudhir Ryan Daniel and Chandrasekharan Rajendran, *"A simulation-based genetic algorithm for inventory optimization in a serial supply chain",* International Transactions in Operational Research, Jan 2005.

10. Truong, T.H. and Azadivar, F., "*Optimal design methodologies for configurations of supply chains*", International Journal of Production Research, Vol. 43, Issue 11, 1 June 2005, 2217–2236.

11. Tong Wu and Peter O'Grady, "*A methodology for improving the design of a supply chain*", International Journal of Computer Integrated Manufacturing, June 2004, Volume17, No. 4, 281- 293.

12. J.K. Cochran and Hung-Nan Chen, "*Generating daily production plans for complex manufacturing facilities using multi-objective genetic algorithms*", International Journal of Production Research, 2002, Vol. 40, No. 16, 4147-4177.

13. Felix T.S. Chan and S.H. Chung, "*A multi-criterion genetic algorithm for order distribution in a demand driven supply chain*", International Journal of Computer Integrated Manufacturing, June 2004, Vol. 17, No. 4, 339-351.

14. P. Pal, C.B.Das, A. Panda and A.K. Bhunia, "*An application of real-coded genetic algorithm (for mixed integer non-linear programming in an optimal two-warehouse inventory policy for deteriorating items with a linear trend in demand and a fixed planning horizon)*", International Journal of Computer Mathematics, Vol. 82, No. 2, Feb 2005, 163-175.

15. Hongwei Ding, Lyes Benyoucef and Xiaolan Xie, "*A multiobjective optimization method for strategic sourcing and inventory replenishment*", Proceedings of the 2004 IEEE International Conference on Robotics and Automaton, April 2004.

16. Hongwei Ding, Lyes Benyoucef and Xiaolan Xie, *"A simulation-based optimization method for production-distribution network design"*, Proceedings of the 2004 IEEE International Conference on Systems, Man and Cybernetics.

17. Hongwei Ding, Lyes Benyoucef and Xiaolan Xie, *"A simulation optimization methodology for supplier selection problem"* International Journal of Computer Integrated Manufacturing, Vol. 18, No. 2-3, March-May 2005, 210-224.

18. Elmahi, I., Merzouk, S., Grunder, O., and Elmoudni, A., *"A genetic algorithm approach for the batches delivery optimization in a supply chain"*, Proceedings of the 2004 IEEE International Conference on Networking, Sensing and Control, Mar 21-23, 2004

19. N. Rezg, X. Xie and Y. Mati, *"Joint optimization of preventive maintenance and inventory control in a production line using simulation"*, International Journal of Production Research, May 2004, Vol. 42, No. 10, 2029-2046.

20. Angerhofer, B., and Angelides, M., *"System dynamics modeling in supply chain management: Research review"*, Proceedings of the 2000 Winter Simulation Conference.

21. Towill, D. R., *"Industrial dynamics modeling of supply chains"*, Logistics Information Management, Issue 4, 1996.

22. Riddalls, C. E., Bennett, S., and Tipi, N.S., *"Modeling the dynamics of supply chains"*, International Journal of Systems Science, 2000, Volume 31, No. 8,969-976.

23. Ovalle, R. O., and Marquez, C. A., *"The effectiveness of using e-collaboration tools in the supply chain: An assessment study with system dynamics"*, Journal of Purchasing and Supply Management, 2003, No. 9, 151-163.

24. Minegishi, S. and Thiel, D., *"System dynamics modeling and simulation of a particular food supply chain"*, Simulation Practice and Theory, 8, 2000, 321-339.

25. Ge, Y., Yang, J. B., Proulove, N., and Spring, M., "System dynamics modeling for supply chain management: A case study on a supermarket chain in the UK", International Transactions in Operational Research, 11, 2004, 495-509.

26. Vlachos, D., Georgiadis, P., and Iakovou, E., "A system dynamics model for dynamic capacity planning of remanufacturing in closed-loop supply chains", Computers and Operations Research, In press.

27. Spengler, T., and Schroter, M., *"Strategic management of spare parts in close-loop supply chains – A system dynamics approach",* Interfaces, Nov/Dec 2003, 33, 6.

28. Rojas, O. B., Gardner, B., Naim, M., *"A system dynamic analysis of officer manpower in the merchant marine"*, Maritime Policy Management, 1999, Vol. 26, No. 1, 39-60.

29. Croom, S., Romano, and P., Giannakis, M., *"Supply chain management: an analytical review for critical literature review"*, European Journal of Purchasing and Supply Management, 6, 2000, 67-83.

30. Tan, C. K., *"Supply chain management: practices, concerns and performance issues"*, Journal of Supply Chain Management, Winter 2002, 38, 1.

31. Croom, S., and Giannakis, M., *"Toward the development of a supply chain management paradigm: a conceptual framework"*, Journal of Supply Chain Management, Spring 2004, 40, 2.

32. Shapiro, J., *"Challenges of strategic supply chain planning and modeling", Computers and Chemical Engineering"*, 2004, 28, 855-861.

33. Terzi, S., and Cavalieri, S., *"Simulation in the supply chain context: a survey"*, Computers in Industry, 2004, 53, 3-16.

34. Talluri, S., and Baker, R.C., "A multi-phase mathematical programming approach for effective supply chain design", European Journal of Operational Research, 141, 2002, 544-558.

35. Agrawal, V., Chao, X., and Seshadri, S., *"Dynamic balancing of inventory in supply chains"*, European Journal of Operational Research, 159, 2004, 296-317.

36. Chiang, K. W., and Monahan, E. G., *"Managing inventories in a two-echelon dual-channel supply chain"*, European Journal of Operational Research, 162, 2005, 325-341.

37. Giannoccaro, I., Pontrandolfo, P., and Scozzi, B., "A fuzzy echelon approach for inventory management in supply chains", European Journal of Operational Research, 149, 2003, 185-196.

38. Kapoor, S., Battacharya, K., Buckley, S., Chowdhary, P., Ettl, M., Katircioglu, K., Mauch, E., Phillips, L., *"A technical framework for sense-and-response business management"*, IBM Systems Journal, 2005, 44, 1.

39. J. Dejonckheere, J., Disney, M.S., Lambrecht, R. M., Towill, R. D., *"Measuring and avoiding bullwhip effect: a control theoretic approach"*, European Journal of Operational Research, 147, 2003, 567-590.

40. Jung, Y. J., Blau, G., Pekny, F. J., Reklaitis, V. G., Eversdyk, D., "A simulation based optimization approach to supply chain management under demand uncertainty", Computers and Chemical Engineering, 28, 2004, 2087-2106.

41. Samaddar, S., Nargundkar, S., and Daley, M., *"Inter-organizational information sharing: The role of supply network configuration and partner goal congruence"*, European Journal of Operational Research, In press.

42. Kelle, P., and Akbulut, A., *"The role of ERP tools in supply chain information sharing, cooperation, and cost optimization"*, International Journal of Production Economics, 93-94, 2005, 41-52.

43. Deb Kalyanmoy, *"Multi-objective optimization using evolutionary algorithms"*, John Wiley & Sons, June 27, 2001, ISBN: 047187339X.

44. Goldberg, E. D., "Genetic algorithms in search, optimization, and machine learning", Addison-Wesley Professional, January 1, 1989, ISBN: 0201157675.

45. Mitchell, M., *"An introduction to genetic algorithms"*, The MIT Press, February 6, 1998, ISBN: 0262631857.

46. Koza, R. J., "Survey of genetic algorithms and genetic programming", WESCON, 1995, IEEE.

47. http://www.iitk.ac.in/kangal/

48. Vensim DSS reference supplement, Ventana Systems, Inc.