University of Windsor

Scholarship at UWindsor

Electronic Theses and Dissertations

Theses, Dissertations, and Major Papers

2008

Open source solution approaches to a class of stochastic supply chain problems

Sicheng Chen
University of Windsor

Open Source Solution Approaches to a Class of Stochastic Supply Chain Problems

by

Sicheng Chen

A Thesis

Submitted to the Faculty of Graduate Studies
through Industrial and Manufacturing Systems Engineering
in Partial Fulfillment of the Requirements for
the Degree of Master of Applied Science at the
University of Windsor

Windsor, Ontario, Canada

2008

Your file   Votre référence
ISBN: 978-0-494-47094-7
Our file   Notre référence
ISBN: 978-0-494-47094-7

ABSTRACT

This research proposes a variety of solution approaches to a class of stochastic supply chain problems, with normally distributed demand in a certain period of time in the future. These problems aim to provide the decisions regarding the production levels; supplier selection for raw materials; and optimal order quantity. The typical problem could be formulated as a mixed integer nonlinear program model, and the objective function for maximizing the expected profit is expressed in an integral format. In order to solve the problem, an open source solution package BONMIN is first employed to get the exact optimum result for small scale instances; then according to the specific feature of the problem a tailored nonlinear branch and bound framework is developed for larger scale problems through the introduction of triangular approximation approach and an iterative algorithm. Both open source solvers and commercial solvers are employed to solve the inner problem, and the results to larger scale problems demonstrate the competency of introduced approaches. In addition, two small heuristics are also introduced and the selected results are reported.

DEDICATION

To my parents

ACKNOWLEDGMENTS

I would like to extend my sincere gratitude and appreciation to my supervisor, Dr. Guoqing Zhang for the wonderful knowledge and experiences he shared, the endless help he gave and the inspiring guidance he offered during the entire period of my Master's study. Special appreciations will be given to Dr. Abdo Alfakih and Dr. Michael Wang for their time and invaluable advices throughout the process. I also sincerely appreciate the endless support from Dr. Reza Lashkari and the knowledge he shared during my course work study.

I would not forget my family members, as this thesis would have been impossible without their constant support and encouragement, and especially for Miss Ting Zhang's constant support. Also, I extend gratitude to my friend Mr. Pedram Sahba for his knowledge and help, and Mr. Patrick Rodd for his encouragement and friendship. I also want to extend gratitude to all my friends in Windsor, Diana Tseng, Caroline Stuart, Steven Yang, Meiling Chen, Jason Yang, Helen Zhang, Yi Duan, Yawei Li, Guohong Hu etc. for their friendship.

TABLE OF CONTENTS

ABSTRACT iii

DEDICATION iv

ACKNOWLEDGMENTS v

LIST OF TABLES ix

LIST OF FIGURES x

LIST of ABBREVIATIONS xi

CHAPTER

1. INTRODUCTION 1

1.1 Background 1

1.2 Research Motivation 3

1.3 Research Objective 4

1.4 Thesis Organization 4

2. LITERATURE REVIEW 6

2.1 Stochastic Supply Chain Modeling and Solution Approaches 6

2.1.1 Demand Uncertainties modeled as Discrete Distributions and Solution Approaches 6

2.1.2 Newsboy Model 10

2.2 Supplier Section and Order Lot Sizing 15

2.3 Nonlinear Integer & Mixed Integer Nonlinear Supply Chain Design and Approaches 17

2.4 Open Source Development and Application 20

2.4.1 Open-Source Initiative 20

2.4.2 COIN-OR 22

3. PROBLEM DESCRIPTIONS AND OPEN SOURCE BASED MINLP SOLUTION APPROACH 26

3.1 Problems Descriptions 26

3.1.1 Model No.1 --- The Configuration of a Manufacturing Firm's Supply Network with Multiple Suppliers 27

3.1.2 Model No.2 --- Optimal Acquisition Policy for a Supply Network with Discount Schemes and Uncertain Demands 28

3.1.3 Model No.3 --- Zhang (2007) 29

3.1.3.1 Assumptions 29

3.1.3.2 Parameters and Variables 30

3.1.3.3 Formulation 31

3.1.3.4 Solution Background 33
3.2 Foundation of Open Source Solution Approaches 34
3.2.1 AMPL 35
3.2.2 BONMIN 39
3.3 External Function Based Open Source Approach 40
3.4 Triangular Approximation Open Source Approach 42
3.4.1 Taylor Series based Triangular Approximation Approach: 45
3.4.2 Covering Range Based Triangular Approximation Approach: 47
4. RESULTS AND ANALYSIS TO BONMIN BASED APPROACHES 48
4.1 Results and Comparison of Three Proposed Approaches 48
4.2 Validation of Triangular Approximation Approach 53
5. NONLINEAR BRANCH-AND-BOUND ALGORITHM 57
5.1 Motivation to the Development of B&B Algorithm 57
5.2 Background of Branch and Bound algorithm for MINLP 58
5.2.1 Inner Nonlinear Programming Problems 58
5.2.1.1 Generalized Reduced Gradient (GRG) and CONOPT 59
5.2.1.2 Interior Point Method for Nonlinear Programming 60
5.2.1.3 Iterative Algorithm for Inner Nonlinear Problem 61
5.2.2 Node Selecting Strategy in Nonlinear B&B 61
5.2.3 Branching Strategy in Nonlinear B&B 62
5.2.4 Obtaining Upper Bound for sub-problems 63
5.3 Branch and Cut 64
5.4 Proposed Nonlinear B&B Algorithm 65
5.5 Pseudo-Code 66
6. COMPUTATIONAL EXPERIMENTS OF NONLINEAR B&B ALGORITHM 68
6.1 Validation of Nonlinear B&B algorithm 68
6.2 Comparison of Different Nonlinear Solvers to Inner Quadratic Problem 69
6.3 Experiments on Larger Scale Problems 71
6.3.1 The Methodology of Generating Large Scale Problems 71
6.3.2 Discussion of Results to Larger Scale Problem 73
6.4 Discussion of Complexity of Nonlinear Branch-and-Bound 79
6.5 Idea of Heuristics 80
6.5.1 Heuristic One 80

6.5.2 Heuristic Two 81

7. CONCLUSIONS AND RECOMMENDATIONS 83

7.1 Conclusions 83

7.2 Contributions 84

7.3 Recommendations to Future Research 86

APPENDIX A: Kim's case 1 87

APPENDIX B: Kim's case 2 87

APPENDIX C: Zhang's case 88

APPENDIX D: Results for Kim's case 2 90

APPENDIX E: A sample of the nonlinear B&B output 90

APPENDIX F: Detailed output of nonlinear B&B for testing substances 92

REFERENCES 97

VITA AUCTORIS 101

LIST OF TABLES

Table 1: Main AMPL files ... 36
Table 2: Comparison among different approaches when Q=2000 ... 50
Table 3: Comparison among different approaches when Q=2050 ... 51
Table 4: Comparison between combined approaches and original ones when Q=2000 ... 52
Table 5: Comparison between combined approaches and original ones when Q=2050 ... 52
Table 6: Objective function values for different approaches when Q changes ... 53
Table 7: Validation of triangular approach on Kim's case 1 model ... 55
Table 8: Validation of nonlinear B&B algorithm ... 69
Table 9: Comparison of different nonlinear solvers to inner quadratic problem ... 71
Table 10: Testing Substances ... 72
Table 11: Statistics for s6-p6-c6-seg2 ... 75
Table 12: Statistics for s6-p6-c6-seg3 ... 75
Table 13: Statistics for s10-p10-c10-seg2 ... 76
Table 14: Simplified statistics for the rest of the examples ... 77
Table 15: Comparison of IPOPT and CONOPT as inner solver ... 79
Table 16: Comparison between NBB and Heuristic 1 ... 82
Table 17: Product specification - input requirements g(i,k) ... 87
Table 18: Supply Costs (c(i,j) and q(i)) ... 87
Table 19: Production Information ... 87
Table 20: Product-related Parameters ... 87
Table 21: Input Requirements g(i,k) ... 88
Table 22: Supply Costs c(i,j) ... 88
Table 23: Resource usage of supplier j per unit of component i produced n(i,j) ... 88
Table 24: Capacity of supplier q(j) ... 88
Table 25: Setup cost for order component i from supplier j (ksetup(i,j)) ... 89
Table 26: Unit hold cost for component i from supplier j, (h(i,j)) ... 89
Table 27: Unit prices of component i from supplier j on price segment 1 (c(ijl)) ... 89
Table 28: Unit prices of component i from supplier j on price segment 2 (c(ij2)) ... 89
Table 29: Segments specification ... 90
Table 30: Results for Kim's case 2 using E-CRTO ... 90
Table 31: A, B, C calculation of s10-p10-c10-seg2 ... 92
Table 32: Detailed solution of s11-p5-c5-seg2 ... 93
Table 33: A, B, C calculation of s15-p10-c10-seg2 ... 93
Table 34: Detailed solution of s15-p10-c10-seg2 ... 93
Table 35: Detailed solution of s15-p10-c10-seg3 ... 94
Table 36: Solution of s15-p15-c15-seg2 ... 94
Table 37: A, B, C calculation of s15-p10-c10-seg2 ... 95
Table 38: Solution of s20-p15-c10-seg2 ... 95
Table 39: Solution of s15-p15-c15-seg3 ... 96
Table 40: Solution to s15-p15-c20-seg2 ... 96

LIST OF FIGURES

Figure 1: Technique oriented classification of supplier selection and lot sizing problem 15
Figure 2: AMPL .mod file illustration (SSH snapshot)... 37
Figure 3: AMPL .dat file illustration (SSH Snapshot) 38
Figure 4: AMPL .x file illustration (SSH Snapshot) 39
Figure 5: External function based open source BONMIN approach in AMPL 41
Figure 6: Taylor series based triangular approximation approach 46
Figure 7: Covering range based triangular approximation approach 47
Figure 8: Graphical illustration of table 6 53
Figure 9: Graphical illustration of table 7 55
Figure 10: Graphical illustration of table 30 56

LIST of ABBREVIATIONS

AMPL: **A** **M**athematical **P**rogramming **L**anguage

ASL: **A**mpl **S**olver **L**ibrary

BBB: **B**ONMIN **B**ranch and **B**ound

B&B: **B**ranch and **B**ound

BONMIN: **B**asic **O**pen-source **N**onlinear **M**ixed **IN**teger programming

COIN-OR: **CO**mputational **IN**frastructure for **O**perations **R**esearch

CONOPT: GAMS larger scale nonlinear programming solver

CPLEX: Commercial LP, MIP and QP solver from ILOG In.

CRTO: **C**overing **R**ange **T**riangular Approximation **O**pen Source Approach

EFO: **E**xternal **F**unction **O**pen Source Approach

EOQ: **E**conomic **O**rder **Q**uantity

E-TSTO: Hybrid **E**FO and **TSTO** Approach

E-CRTO: Hybrid **E**FO and **CRTO** Approach

GA: **G**enetic **A**lgorithm

GAMS: **G**eneral **A**lgebraic **M**odeling **S**ystem

GRG: **G**eneralized **R**educed **G**radient

INFORMS: **IN**stitute **F**or **O**perations **R**esearch and **M**anagement **S**cience

IPOPT: **I**nterior **P**oint **OPT**imizer

LP: **L**inear **P**rogramming

MINOS: A linear and nonlinear mathematical optimization solver

MIP: **M**ixed **I**nteger **P**rogramming

MIQCP: **M**ixed **I**nteger **Q**uadratic **C**onstrained **P**rogramming

NBB: **N**onlinear **B**ranch and **B**ound

O. B. Value: **OB**jective **V**alue

OSI: **O**pen **S**ource **I**nitiative

QP: **Q**uadratic **P**rogramming

SBB: **S**imple **B**ranch and **B**ound, GAMS MINLP solver

SQP: **S**equential **Q**uadratic **P**rogramming

TSTO: **T**aylor **S**eries **T**riangular Approximation **O**pen Source Approach

XPRESS: GAMS nonlinear and quadratic solver

1. INTRODUCTION

1.1 Background

The increasing competitive pressures in the global marketplace, coupled with the rapid advances in information technology have brought supply chain planning into the forefront of the business practices for most manufacturing and service organizations.

"Supply chain management can be viewed as both an emergent field of practice and an emerging academic domain. Neither perspective is fully mature but each has considerable promise" (Storey et al., 2006). This conclusion is made by critically assessing current developments in the theory and practice of supply chain management, and through such assessments to identify barriers, possibilities, and key trends. It is pointed out that much of the theory in supply chain management is based on idealized schemes of optimal routes and quantities for demand fulfillment when it is considered from a whole-network or chain perspective. They compiled a picture of current supply chain practices and have identified a number of organizational and behavioral barriers to the realization of the more idealistic depictions of the seamless and end to end chain that should be responding to customer demand. The following lists the core component ideas:

1. Seamless flow from initial source to final customer
2. Demand-led supply chain (only produce what is pulled through)
3. Share information across the whole chain (end to end pipeline visibility)
4. Collaboration and partnership (mutual gains and added value for all; win-win; joint learning and joint design and development)
5. IT enabled
6. All products direct to shelf

7. Batch/pack size configured to rate of sale
8. Customer responsive
9. Agile and lean
10. Mass customization
11. Market segmentation

What is worth mentioning is that the academic disciplines normally have core sets of concerns or problems, the variability and uncertainty within supply chain management are the core problems in both academic area and real world engineering application nowadays. From the second component listed above, we know that demand-driven supply chain is critical for the overall success. However, the demand uncertainty brings difficulties in terms of both supply chain modeling and model solving process in academic research community.

There has been a large amount of literatures addressing demand uncertainties in supply chain management. The foremost consideration in incorporating uncertainties into the planning decisions is the determination of the appropriate representation of the uncertain parameters (Gupta 2003). Two distinct methodologies for representing uncertainty can be identified, which are the scenario-based approach and the distribution-based approach. In the former approach, the uncertainty is described by a set of discrete scenarios capturing how the uncertainty might play out in the future with each scenario assigned a probability of its future occurrence. However, the applicability of the scenario based approach is limited by the fact that it requires forecasting for all possible outcomes of the uncertain parameters. In cases where a natural set of discrete scenarios cannot be identified and only a continuous range of potential futures can be predicted, the distribution- based approach is used by assigning a probability distribution to the continuous range of potential outcomes. The normality assumption is widely invoked in literature (Wellons and Reklaitis, 1989; Nahmias, 1989) as it captures the essential features of demand uncertainty and is convenient to use (Gupta, 2003). Nevertheless, the normality assumption formulation makes it hard to find an efficient solution approach in many cases, particularly, when it is combined with other considerations for in constructing models.

1.2 Research Motivation

Nowadays, the approaches and software for stochastic supply chain problems with discrete distribution functions are common in both the commercial world and the academic world; however, the cases with continuous distribution demands are less explored and so as good stochastic software for those problems. Among the same class of problems that specifically deal with supplier selection and order quantity determination decisions facing continuously distributed demands of finished products, the work of Kim et al. (2002), Zhang and Ma (2007) and Zhang (2007) are the most representative ones, among which, particularly, the Zhang (2007) model is the most comprehensive model. This model developed a MINLP model by combining strategic acquisition decisions with inventory management, where the manufacturer produces multiple products facing uncertain demand which is assumed to be normal distribution. Each product needs certain amounts of raw materials to be produced, which can be purchased from different suppliers with a quantity discount for different amounts and is going to be bought in a single planning period, such as a year. A MINLP model was build and solved using GAMS and its MINLP solver. However, due to the external functions introduced in the commercial software, the solving process exhibited super sensitivity in terms of parameter changes, not to mention larger scale problems.

Open source initiatives have been prevailing in the operations research community over the years. Their aim is to provide an open platform where both the source codes and algorithms are available to researchers and which also makes it possible for different researchers to compare their own algorithms for certain problems under relatively fair circumstances. For this consideration, we are determined to get more accessible and controllable solution approaches to this class of problems, especially for the model in Zhang (2007) based on open source solver, which can deliver more robust and reliable solutions, especially for larger scale problems.

1.3 Research Objective

The objective is to deliver more accessible, reliable, controllable and efficient solutions to the current dimension, and more importantly for larger scale problems based on the model in Zhang (2007). Therefore, we will demonstrate the usage of the open source software package BONMIN, and AMPL which serves as a comprehensive and powerful algebraic modeling language. In addition, an independent branch-and-bound algorithm, which is tailored to the model and combined with open-source solver, will be implemented in GAMS to reveal the insight of the solution approach and compare it with other available approaches.

1.4 Thesis Organization

The thesis is organized as follows: In Chapter II the literatures relevant to this research will be introduced and subsequently in Chapter III, the class of problems we are dealing with will be described, and the Open Source based MINLP package Solution Approaches are proposed. Chapter IV provides some preliminary results to the example problem and validates the proposed approaches through different experimental tests and comparisons between the different approaches that were used previously. Chapter V delivers the proposed branch and bound algorithm framework specifically for the model in Zhang (2007). Chapter VI addresses the analysis of the larger scale problems and reports two heuristics. Finally, conclusions and recommendations are made and possible future improvements are discussed.

2. LITERATURE REVIEW

There has never been a shortage of general supply chain modeling literatures since the debut of the Supply Chain Management. Different aspects of Supply Chain Management such as the new design and methodology, modeling and analysis, the concept and implementation in real world industry are intensively covered in the literatures.

In this chapter, different themes of literatures have been organized into the following sections: Supplier Selection and Order Lot Sizing, Stochastic Supply Chain Design and solution approaches, Nonlinear Integer & Mixed Integer Nonlinear Supply Chain Design and Approaches, Open-Source development and application.

2.1 Stochastic Supply Chain Modeling and Solution Approaches

As discussed in the introduction section, the considerations of stochastic factors in supply chain modeling are prevailing nowadays, which reveals the nature of the business world. Various stochastic factors or uncertainties can be indentified in the business systems or supply chain systems and the existence of uncertainty factors brings much complexity for both modeling and computational process. One of the most important criterions to evaluate a good supply chain network is demand-driven, and demand itself is one of the key sources of uncertainties in any production-distribution system. The following literatures are organized in a way that demand uncertainties are classified into discrete and continuous distributions.

2.1.1 Demand Uncertainties modeled as Discrete Distributions and Solution Approaches

Santoso et al. (2005) proposed a stochastic programming model and solution algorithm for solving supply chain network design problems of a realistic scale. Their solution integrates the sample average approximation (SAA) scheme, with an accelerated Benders decomposition algorithm to quickly compute high quality solutions to large-scale stochastic supply chain design problems with a huge (potentially infinite) number of scenarios. The proposed algorithm was proved robust by two supply chain network problems. The big contribution of this paper is that the new algorithm they proposed is able to tackle multistage stochastic problems with huge number of scenarios that is normal case in the real world.

Leung and Wu (2005) developed a two-stage stochastic programming model to solve border-crossing distribution problems in an environment of uncertainty. They followed the classic two stage stochastic programming with recourse formulation strategy to formulate the problem. Under different economic growth scenarios with various possibilities, an equivalent mixed integer linear deterministic model was development, and solved by LINDO package. In addition, the subsequent analysis was done in terms of the changes to the possibility associated with each scenario. The authors developed a robust model; however, the solution approach to the problem is not representative enough for other two stages or multi-stage stochastic optimization problems.

Lucas et al. (2001) presented a two-stage resource allocation model with 0-1 discrete variables. Using scenarios to represent the uncertainties in demand, they built a very large scale mixed integer-programming problem. In order solve this intractable problem, the Lagrange relaxation and its corresponding decomposition of the initial problem was employed to approximate the given problem where Lagrange relaxation is reinterpreted as a column generator. Their approach avoided the complicated effort on the intractable large scale MIP problem and has been applied to study supply chain capacity investment and planning.

Albornoz et al. (2004) proposed the way to obtain an optimum policy in the capacity expansion planning of a particular thermal-electric power system by formulating a two-stage stochastic integer programming. The existent uncertainty related to the future availability of the thermal plants currently under operation. They used the so-called L-shaped method to solve the problem numerically. AMPL was employed as the modeling platform of the problem coupled with CPLEX as the solver to implement the algorithm. In this paper, they proposed a good methodology for the stochastic integer problem.

Alonso-Ayuso et al. (2003) presented a two-stage 0-1 stochastic modeling and a related algorithmic approach for a Supply Chain Management problem under scenario based uncertainties. In the model, strategic decisions were made in the first stage and the second stage was included by the tactical decisions. A splitting variable mathematical representation via scenario was presented for the stochastic version of the model and a two-stage version of a Branch and Fix Coordination (BFC) algorithmic approach was proposed to solve the stochastic 0-1 program. This paper provides a benchmark solution approach for multi-stage stochastic integer programming problems and is invoked frequently in the subsequent research.

Gupta et al. (2000) utilized the framework of mid-term, multisite supply chain planning under demand uncertainty to try to avoid both inventory depletion at the production sites and excessive shortage at the customer. A chance constraint programming approach combined with a two-stage stochastic programming methodology was presented to capture the trade-off between customer demand satisfaction and production costs. In the proposed model, the production decisions are made before demand realization while the supply chain decisions are delayed. The challenge associated with obtaining the second stage recourse function is resolved by first obtaining a closed-form solution of the inner optimization problem using linear programming duality followed by expectation evaluation by analytical integration. In addition, analytical expressions for the mean and standard deviation of the inventory are derived and used for setting the appropriate customer demand satisfaction levels in the supply chain.

Gupta and Maranas (2000) proposed a two-stage stochastic programming approach to incorporate demand uncertainty in multisite midterm supply-chain planning problems. Under their methodology, the supply chain decision will not be made until the production decision is made. They obtained the closed-form solution of the inner optimization problem using linear programming duality. The evaluation of the expected second stage costs is achieved by analytical integration yielding an equivalent convex mixed-integer nonlinear problem (MINLP). Compared with Monte Carlo sampling, their computational effort is much smaller.

Ahmed et al. (2003) addresses a multi-period investment model for capacity expansion in an uncertain environment, such as uncertain demand and cost parameters which was modeled using a scenario tree approach. A multi-stage stochastic integer programming formulation for the problem was developed and a reformulation of the problem was proposed using variable disaggregation to exploit the lot-sizing substructure of the problem. The reformulation approach dramatically reduces the LP relaxation gap of larger scale integer program and a heuristic approach was also presented produce good quality integer solutions. The authors finally combined a branch and bound algorithm that makes use of the reformulation strategy as a lower bounding scheme, and the heuristic as an upper bounding scheme, to solve the problem to global optimality.

For comprehensive structural properties of and algorithms for stochastic integer programming models, please refer to Haneveld and Vlerk, (1999).

In this research, the solution approach is focused on a supply chain problem with demand uncertainty of continuous distribution which shares many similarities with the formulation of Newsboy model. This indicates that certain solution approach adopted in the literatures of Newsboy or Newsvendor model could provide some insights to the class of the models. The followings provide the corresponding review.

2.1.2 Newsboy Model

Another big class of stochastic supply chain problems is Newsboy Problem, which is also known as Newsvendor Problem or Newsstand Problem. The newsboy problem as the single period stochastic inventory model is found to be a suitable tool for decision-making regarding stocking issues in today's supply chains. Simply described, the Newsboy problem deals with situations where the demand for a commodity is uncertain (random) and those items that are ordered but remain unsold or unused at the end of the cycle become obsolete (Abdel-Malek et al., 2004).

Rekik et al. (2004) analyzed a single-period, uncertain demand inventory model under the assumption that the quantity ordered (produced) is a random variable. They conducted a comprehensive analysis of the well known single period production/inventory model with random yield. Under the hypothesis that demand and the error in the quantity received from supplier are uniformly distributed, they obtained closed-form analytical solutions for all values of parameters. They also provided the analysis of the benefit achieved by eliminating errors.

Abdel-Malek and Montanari (2004) did a landmark work in constrained Newsboy problem where they proposed an exact solution procedure for the formulation with the uniform the demand probability density function. Further more, a Generic Iterative Method (GIM), which yields optimum, or near optimum, solution for general continuous density functions of the demand, was illustrated to make it possible for one to stop the computation when the desired level of accuracy is achieved. Subsequently, Abdel-Malek and Montanari (2005) extended the constrained Newsboy problem to the scenario where infeasible ordering quantities (negative) were obtained when applying the solution technique in Abdel-Malek et al. (2004). The solution space was divided into three regions according to different availability of the budget and numerical examples were also given to illustrate the application of the developed procedures.

Again, In Abdel-Malek and Montanari (2005), they developed a methodology to examine the dual of the solution space of constrained multi-product newsboy problem with two constraints and propose an approach to obtain the optimum batch size of each product. The approach is based on utilizing the Lagrangian Multipliers, Leibniz Rule, Kuhn–Tucker conditions to obtain the optimum or near optimum solution combined with iterative techniques whenever necessary.

Weng (2004) developed a generalized newsvendor model to analyze the coordinated quantity decisions between the manufacturer and the buyer. The manufacturer and the buyer operate to meet random demand of one product with a short lifecycle. The main contribution for them is to generate an analytical result for the model and the analytical process also yields the insights into the coordination structure between the manufacturer and the buyer. The quantity discount policy combined with Newsvendor problem provides some real world insights to the application of Newsvendor problem.

Geunes et al. (2001) address inventory decisions in the context of the reorder-point, order-quantity policy in infinite-horizon, stochastic lead-time demand inventory systems in which the parameters may be non-stationary. They also developed a heuristic based on a simple EOQ model and one-period newsvendor model. The heuristic approach is also proved to be nearly as well as the optimal policy derived from complicated mathematical procedures. But only single product was considered in their model and the heuristic approach was also based on single product and single supplier.

Yang et al. (2007) studied a single-product multi-supplier selection problem, where the product faces uncertain demand and different suppliers face different yield and prices. The buyer has to decide whether or not to order from each supplier, and if so how much to order given the stochastic demand nature of the products. A solution method based on a combination of the active set method and the Newton search procedure was proposed and the computational study also showed the effectiveness of their algorithm.

Rekik et al. (2006) considered a single-period, uncertain demand inventory model under the assumption that the quantity ordered is a random variable. Based on a comprehensive analysis of the well known single period production and inventory model with random yield, they extend some of the results existing in literature to a scenario where demand

and the error in the quantity received from supplier are uniformly distributed. A closed form analytical solution approach was provided and an analysis under normally distributed demand and error was also provided.

Niederhoff (2007) provided an approximating programming technique to solve the multi-product and multi-constraint newsvendor problem. In stead of Lagrange Relaxation Techniques employed by the literatures before, the author took advantage the separable nature of the problem and developed a close approximation of the optimal solution using convex separable programming for any demand distribution in the traditional newsvendor model and its extensions. Since their approach is totally independent of the Lagrange Multiplier based methodologies, it makes it possible to extend the newsboy model to a new level.

Areeratchakul and Abdel-Malek (2006) proposed a solution approach for the multi-product newsvendor model with multiple constraints. The methodology was based on quadratic programming and triangular presentation of the area under cumulative probability distribution function of the demand. Their approach could provide exact solutions for uniform distribution and satisfactory approximations to other distribution functions such as normal distribution or exponential distribution.

Khouja and Mehrez (1996) extended the classic newsvendor problem to the situation where the decision makers face a multi-product newsboy problem in which multiple discounts are used to sell excess inventory under a storage or budget constraint. A Lagrange Multiplier based algorithm was developed for the problem and the numerical example was also demonstrated to prove the effectiveness of the algorithm.

Lau and Lau (1995, 1996) introduced a Lagrange based numerical method to solve the multi-product multi-constraint newsboy problem. The main idea of the problem is that the proposed approach requires first obtaining the solution for the unconstrained model in order to initiate their numerical procedures.

Vairaktarakis (2000) presented robust newsboy models with uncertain demand. Instead of describing uncertainty by means of probability density functions, the author described uncertainty using two types of demand scenarios, namely interval and discrete scenarios. For interval demand scenarios they only require a lower and an upper bound for the uncertain demand of each item, while for discrete demand scenarios they require a set of likely demand outcomes for each item. Using the above scenarios to describe demand uncertainty, they develop several mini-max regret formulations for the multi-item newsboy problem with a budget constraint. For the problems involving interval demand scenarios, linear time optimal algorithms were developed and for the models with discrete demand scenarios, they were solvable by dynamic programming. The model was also extended to the mixture of the above two scenarios mentioned above.

Erlebacher (2000) has addressed the model of the capacitated multi-item newsvendor problem in cases where the cost structure is similar. He developed exact and heuristic solutions depending on the types of the demand probability distribution functions for different products.

As discussed in the publications above, there have been a tremendous amount of publications regarding newsvendor problem and its various forms of extensions; however, most of the techniques shared in the publications are based on the unconstrained optimum of the original classical newsboy problem. The problem formulated by Zhang (2007) shares some similarities with newsvendor model but also bears many discrepancies regarding both definition and formulation of the problem.

2.2 Supplier Section and Order Lot Sizing

With the advent of supply chain management, much attention is now devoted to supplier selection. The decision that is needed to make in supplier selection and order lot sizing problem can be categorized as follows (Aissaoui et al. 2007):

- What product to order?
- In what quantities and from which suppliers?
- In which periods?

From the perspective of technique oriented classification, the published worked can be divided into two major groups: single objective group and multi-objective group. See Figure 1 (Aissaoui et al. 2007) for detail.

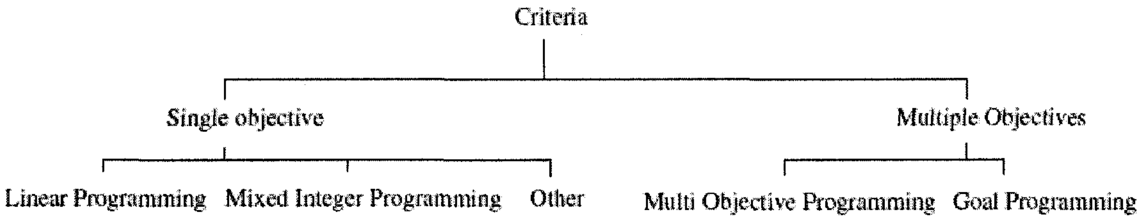


Figure 1: Technique oriented classification of supplier selection and lot sizing problem

Problems modeled as other programming in single objective category in Figure 1 includes:

- Dynamic programming: Fabian et al. (1959), Morris (1959), Kingsman [1986].
- Nonlinear programming: Pirkul and Aras (1985), Hong and Hayya (1992), Rosenblatt et al. (1998).

- Stochastic programming: Bonser and Wu (2001).
- Decision theory: Ammer (1968).

Stadtler (2007) developed a generalized mixed integer linear programming model which considers not only the all-units discount but also the incremental discount case which was only tackled by various ways of heuristics. The objective function in the proposed model is chosen in a way that resolves conflicts among proponents of a purely cost oriented and a cash flow oriented modeling approach. This paper also provided a review of the available research on quantity discount scheme.

Basnet and Leung (2005) extended the single product, multi-period inventory lot-sizing model into multiple products and multiple suppliers and the demand of multiple discrete products is known over a planning horizon. An enumerative search algorithm and a heuristic were presented to solve the problem.

Minner (2003) provided comprehensive reviews in inventory models with multiple supply options and discussed their contribution to supply chain management. Inventory models which include several suppliers in order to avoid or reduce the effects of shortage situations were discussed in the paper and the author also presented the related inventory problems from the fields of reverse logistics and multi-echelon systems. Combining Aissaoui et al. (2007) and Minner (2003) provides different discount schemes and multi-supplier selection strategies under inventory models. Moreover, many effective algorithm approaches were also proposed in the literatures invoked and classified in the two papers mentioned previously, which are good resorts for the researcher who are interested in this topic.

2.3 Nonlinear Integer & Mixed Integer Nonlinear Supply Chain Design and Approaches

Since this research deals with the solution approach of a stochastic supply chain problem modeled as a MINLP, it is important to review the solution approaches of the MINLP supply chain problems.

Ko and Evans (2007) considered a supply chain management scenario from a perspective where the third party logistics providers (3PLs) must operate supply chains for a number of different clients who want to improve their logistics operations for both forward and reverse flows under the current globally fierce competition. Different from the past, during which the design of distribution networks has been independently conducted with respect to forward and reverse flows, a dynamic integrated distribution network to account for the integrated aspect of optimizing the forward and return network simultaneously was developed and modeled as MINLP. Due to the complexity of the problem as NP-hard, a genetic algorithm-based heuristic with associated numerical results is presented and a base-line example was tested by the genetic algorithm approach. Moreover, in order to assess the computational effectiveness of the GA, the original mathematical model was converted into a linear model through the use of dummy variables and additional constraints owing to the nonlinear components in the objective function. Finally, the results of the linear model, which were obtained from LINGO, were compared with those from GA approach and they conclude that GA based heuristic algorithm is more suitable for larger scale problems.

Torabi et al. (2006) proposed a model which investigates the lot and delivery scheduling problem in a simple supply chain where a single supplier produces multiple components on a flexible flow line (FFL) and delivers them directly to an assembly facility (AF). The main objective is to find a lot and delivery schedule that would minimize the average of holding, setup, and transportation costs per unit time for the supply chain. A MINLP model was created to represent the problem. Based on the special characteristics of the problem, an optimal enumeration method was developed to guarantee the optimal of the problem. However, in order to tackle the medium or larger scale problems, a hybrid genetic algorithm (HGA) was also developed, which incorporates a neighborhood search (NS) into a basic genetic algorithm that enables the algorithm to perform genetic search over the subspace of local optima. Some results were also shown in the paper to demonstrate the promising performance of HGA.

Wang and Sarker (2004) studied a single-stage supply chain system that is controlled by kanban mechanism, which was pioneered by Toyota Motor Company in Japan and subsequently it was adopted by numerous other Japanese and US companies for applying the just-in-time manufacturing principles. The whole Kanban system was finally modeled as a MINLP which was solved optimally by branch-and-bound method to determine the number of kanbans, batch size, number of batches, and the total quantity over one period and a logistics system for controlling the production as well as the supply chain system is developed, which results in minimizing the total cost of the supply chain system. However, since the number of integers increases with the number of kanban stages, for multi-stage supply chain systems, the computational solution time of B&B is of concern, and the limitation for the size of problems that can be solved optimally by B&B was not studied.

Wang and Sarker (2006) extended the work of Wang and Sarker (2004) to a multi-stage supply chain system that operates under a JIT (just-in-time) delivery policy. Again, for a multi-stage supply chain system, a mixed-integer nonlinear programming (MINLP) problem was formulated to determine the number of kanbans, the batch size, etc. Similarly, it is solved optimally by branch and bound method, moreover, a greedy heuristic to avoid the large computational time in branch-and-bound algorithm is developed for solving a large MINLP. This paper provides an extremely good comparison among the solution procedures between exact algorithms and heuristic algorithms.

Lieckens and Vandaele (2007) concerned with the efficient design of a reverse logistics network. Based on the traditional models formulated as mixed integer linear programs (MILP-model), they extended the model to the scenario where the queuing model was combined to account for the high degree of uncertainty inherent to reverse logistics networks, such as lead time and inventory positions. Due to the nonlinear relationships, a MINLP model was presented for a single product-single-level network and a differential evolution technique based genetic algorithm was developed to solve an example problem. However, no larger scale problems were solved in this paper. Interestingly, they divided the methods to solve MINLPs into two major categories: deterministic and stochastic. Deterministic methods have, such as branch-and-reduce and the BB branch-and-bound, interval analysis-based methods, etc, featuring running through the algorithm in a predefined way, which means that the next step of computation is exactly determined. While stochastic global optimization methods (also known as adaptive random search methods) such as differential evolution and adaptive Lagrange-multiplier methods, etc, run through the algorithm in a random way, which means that the next step of computation is undetermined.

2.4 Open Source Development and Application

2.4.1 Open-Source Initiative

Open source is a phenomenon in computer science that is increasingly receiving attention in the popular press. The underlying philosophy of open source is to promote software reliability and quality by supporting independent peer-review and rapid evolution of source code. This philosophy is pragmatically advanced by using copyright law in a nontraditional way (Ladanyi et al. 2005).

An open source license implies that the software that it covers should have its source code included with its package. However, there are other policies that open source software must follow, and these are all included in the Open Source Initiative (www.opensource.org). All open-source licenses certified by the Open Source Initiative adhere to principles set forth in the Open Source Definition. The Open Source Definition version 1.9 states criteria on nine fundamental issues, including access to source code, free distribution, and prohibition of discrimination. However, it needs to be clear of the difference between public domain and open source, unlike the public domain software such as freeware or shareware; open source software is clearly copyrighted. Once software gets certified by the open source initiative, it must permit the unrestricted redistribution of source code, which does not discriminate so that diversity and participation are maximized. For the comprehensive information regarding open source initiative and several of certified open source software, please refer to www.opensource.org official website.

It is also necessary to mention the unique development paradigm of developing complex software which is high-performance, robust and secure. For a successful open source project (e.g. Linux Operating System or COIN-OR), a virtual community of volunteer developers spontaneously arises from among users who may download the source code, try it on their own purposes and make modifications and may also may find and fix bugs, extend functionality, and port to new platforms. Eventually, relatively large number of developers working simultaneously, the code evolves rapidly (Ladanyi et al. 2005). By opening source code for peer review and rapid evolution under an open-source license, computational results can be reproduced, fair comparisons of algorithm performance can be made, the best implementations can be archived and built on, code reinvention can be minimized, implementation innovation knowledge can be transferred, and collaboration and software standards can be fostered (Lougee-Heimer, 2002).

Open-Source software bears the following advantages which are critical for this research:

- Researchers can read, redistribute, and modify the source code.
- Researchers can improve it, adapt it, and fix bugs by trying different problems.
- It has greater availability and flexibility over commercial software.

With the clear understanding of the upside of open source, we know that for an individual researcher, perhaps the most noticeable negative consequence of the current OR research-software practices is the need to re-make pre-existing software. New algorithmic ideas are frequently tested by computationally benchmarking them against published techniques. To make a comparison meaningful, the competing implementations need to be run in the same computing environment over the same test sets. Open source environment in Operations Research Community makes it possible to do so.

However, open source presents an attractive alternative for the OR community, it is by no means a panacea. Operations research is a comparatively specialized area, the number of developers is correspondingly smaller and Writing software for peer review (let alone peer-extension and peer-maintenance), can be a non-incremental effort as compared to writing software for one's own use (Ladanyi et al. 2005).

Linux-alike system appears as the most well known and successful example in open source development. In this research, we focus on the implementation of Open-Source initiative in Optimization and Operations Research (COIN-OR) on the supply chain problem modeled as MINLP.

2.4.2 COIN-OR

The COIN-OR (Computational Infrastructure for Operations Research) project is an initiative to spur the development of open-source software for the operations research community and is an initiative to promote open-source software resources for operations research professionals. The idea for the initiative was conceived by IBM Research. The goal was to create a community-owned, community-operated repository of open source software to meet the needs of OR professionals. It was hosted by IBM from August 2000 and then INFORMS board unanimously voted to become the new host of COIN-OR in November 2002 .Success was defined as having COIN-OR become community-owned and community-operated (www.coin-or.org).

Under the framework of COIN-OR, a variety of software tools are under development by a heterogeneous group of volunteers from industry, academia, and government. There are different modules for linear programming, integer programming, nonlinear programming, subgradient optimization, and tabu search in the source code repository at http://www.coin-or.org. Currently all the modules are under the OSI-certified Common Public License (CPL). Having all the different software projects under the same license allows users to mix-and-match code without having to worry about creating licensing nightmares.

Currently, the main available components of COIN-OR are:

- BCP - Branch-Cut-Price Framework;
- CBC - COIN-OR's native branch and cut code;
- CGL - Cut Generator Library;
- CLP - (COIN-OR LP) a Simplex solver;
- COPS - COIN-OR Open Parallel Search Framework;
- IPOPT - Large-Scale Nonlinear Optimization;
- NLPAPI - a library for defining nonlinear programming problems;
- OSI - Open Solver Interface;
- OTS - Open Source Framework for Tabu Search;
- SMI - Stochastic Modeling Interface;
- Bonmin - Basic Open-source Nonlinear Mixed INteger programming, an experimental open-source C++ code for solving general MINLP problems;
- LaGO - Lagrangian Global Optimizer, for the global optimization of non-convex mixed-integer nonlinear programs;

- GAMSlinks - GAMS/COIN-OR Links, links between GAMS (General Algebraic Modeling System) and solvers that are hosted at COIN-OR;

Some critical modules, which are still evolving, are illustrated in detail as following:

The *Open Solver Interface* (OSI) is an API coded in C++, which enables implementations to be "solver agnostic" (Lougee-Heimer, 2002). OSI make it possible for different algorithms to be implemented and then run using any solver having an Open Solver Interface instantiation with no additional effort by providing an abstract base class to a generic linear programming (LP) solver, along with derived classes for specific solvers. Many applications may be able to use the OSI to insulate them from a specific LP solver. Currently, interfaces to both commercial solvers (e.g. ILOG CPLEX, IBM OSL, and XPRESS-MP) and open source solvers (e.g. CBC, CLP) are available. See https://projects.coin-or.org/Osi for detail.

BONMIN, as the main module that has been dealt with in this research, it is necessary to give a basic review here and I will discuss the detailed application in the later chapter. BONMIN as an open source MINLP module developed in C++ is a collaborated effort between IBM Corporation and Carnegie Mellon University. It incorporates several already-existing open source packages (Clp, Cbc, Cgl, CoinUtils, Ipopt, Osi) and third party software (Ampl Solver Library, Blas, CPLEX, Lapack), and can be operated under both Windows and Unix-alike systems. Bonmin feathers the following algorithms (https://www.coin-or.org/Bonmin):

- B-BB, a NLP-based branch-and-bound algorithm
- B-QA, an outer approximation decomposition algorithm

- B-QG, an implementation Quesada and Grossmann's branch-and-cut algorithm
- B-Hyb, a hybrid outer approximation based branch-and-cut algorithm

These algorithms are exact when the objective and constraints are convex functions, but in case at least one of objective and constraints functions is non-convex, the algorithms give heuristic solutions (Bonami et al. 2005).

In part of this research, the focus will be on the framework of BONMIN package to solve the stochastic supply chain problem. The detailed methodology using BONMIN in this project will be discussed in the following chapters. Information on the other tools is available on the COIN-OR project web site at http://www.coin-or.org.

3. PROBLEM DESCRIPTIONS AND OPEN SOURCE BASED MINLP SOLUTION APPROACH

3.1 Problems Descriptions

In today's global economy, competitive advantage gained from a successful manufacturing strategy does not guarantee success. In order to succeed globally, it is becoming very important for companies to have appropriate supply chain strategies in addition to appropriate manufacturing strategies. Supplier selection is an important strategic or tactical level decision in the current economic climate where outsourcing has been a prevailing situation in the business world, due to the rocketing of the low-cost economic super powers, such as China and India. Provided with the decision regarding supplier selection, it results in the decision as to how much to order from each of the suppliers or outsourcers to maximize the profit or minimize the cost within a certain period of time. Fortunately, as mentioned in the literature reviews, the majority of the supplier selection and lot sizing models are linear programs or Mixed Integer Linear Programming problems, which can be comparatively easy to solve by currently available solvers. The efficient supplier chain should be a demand driven chain (Storey et al. 2006), as when the demand becomes stochastic in stead of deterministic, the problem becomes more complicated and the computational effort needed to solve the model also increases substantially. Stochastic demand can be represented in two ways: discrete and continuous distributions. Continuous distributions normally require more computational efforts compared with the discrete distributions. When these factors are combined with inventory control, which is sometimes nonlinear part, the problems becomes even more complicated.

This class of problems being addressed shares a similar format of revenue generating formulation as newsboy or newsvendor models. However, these problems also bear much more complicated considerations in capacity constraints, moreover, since the remanufacturing process of the components is added on top of the pure buying and selling --- retailing, more decision variables are created than newsvendor model. The involvement of supplier selection makes binary variables a must to formulate a Mixed Integer problem and other considerations of inventory makes it possible for it to be more complicated as a nonlinear problem. The more factors that are considered, the more complete the model is and therefore the more computing efforts are expected. Generally, the following works represent this class of problems as introduced from 3.1.1 to 3.1.3

3.1.1 Model No.1 --- The Configuration of a Manufacturing Firm's Supply Network with Multiple Suppliers

The model described in Kim et al. (2002) considers a supply chain network consisting of a manufacturer and its suppliers, where each product of the manufacturer is composed of several components which are purchased or outsourced among different suppliers. The model was formulated with a similar objective function to the newsvendor problem with linear constraints in terms of both manufacturing capacity of manufacturers and suppliers. The formulation details are as follows:

Maximize

$$\sum_{k=1}^{K}\left\{\int_0^{y_k}[r_k z_k - b_k(y_k - z_k)]f(z_k)dz_k + \int_{y_k}^{\infty}[r_k y_k - a_k(z_k - y_k)]f(z_k)dz_k\right\} -$$

$$\sum_{k=1}^{p} d_k y_k - \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}x_{ij}$$

Subject to Capacity Constraints:

$$\sum_{k=1}^{p} b_{i,k}y_k \leq \sum_{j=1}^{m} x_{ij}\ , i = 1, \dots, n,$$

$$\sum_{i=1}^{n} v_{i,j} x_{i,j} \leq q_j \,, \ j = 1, \ldots, m,$$

$$\sum_{k=1}^{K} t_k y_k \leq Q \,,$$

$$x_{i,j} \,, y_k \geq 0,$$

Similar parameters and variables definition can be found in section 3.1.3. The problem was solved by an iterative algorithm in Kim's paper using a Lagrange multiplier method, which is a shared methodology for many multi-constraints newsvendor problems and guarantees global optimum for the problem. This problem presents a basic model, as is seen in section 3.1.2 and 3.1.3, as the main solving difficulty involved is the continuously distributed demand, which incurs the integration formulation found in the objective function. The model is solved and verified by the proposed approach in chapter 4.

3.1.2 Model No.2 --- Optimal Acquisition Policy for a Supply Network with Discount Schemes and Uncertain Demands

Zhang and Ma (2007) proposed a discount scheme, with different amounts of raw materials to be purchased from various suppliers based on the model in Kim et al. (2002) with inclusion of binary variables. The formulation of the problem is exactly the same as the problem in section 3.1.3 after inventory considerations are removed from the model. The problem was implemented in GAMS coupled with C coded external function. The MINLP model was solved by SBB solver with different trials of NLP solvers such as CONOPT and MINOS. However, unlike the guaranteed optimal solution approach given in Kim et al (2002), given the black-box nature of GAMS's solver, results in not being convincing enough that the solution of GAMS is optimal and GAMS's solvers are not available to all researchers as it is a commercial solver.

3.1.3 Model No.3 --- Zhang (2007)

The model developed in Zhang (2007) combined model No.1 and No.2, inventory cost was also considered in the model which incurs an additional nonlinear portion found in the objective function. Obviously, problem No.3 is the most representative and complex problem in this class Therefore, it is necessary to make a complete description of the model (Zhang, 2007).

3.1.3.1 Assumptions

- A two-tier supply-manufacturing chain problem is considered.
- One cycle of the manufacturer's long-term production period, which is normally a year, is considered. The decision process deals with both a long-term planning problem that explores supplier selection for each material, and raw materials purchasing over the production cycle from each supplier, as well as a short term planning problem that suggests how often to place orders with each supplier.
- Inventory management costs such as holding and ordering cost are also considered.
- Both shortage and overage costs are allowed at the manufacturer's site with the cost of a_k and b_k per unit of product k respectively.
- Uncertain product demands follow normal distributions with different parameters. Product k has demand z_k, probability density function $f(z_k)$, mean μ_k and standard deviation σ_k.

- Suppliers offer all-unit quantity discounts for purchases above certain quantities, which vary depending on the size of the order for one single product from one supplier.

3.1.3.2 Parameters and Variables

The followings are parameters in the model:

e_k : Unit production cost for product k,

r_k : Unit sales revenue of product k,

m_{ik} : The amount of raw material i required for one unit of product k,

$n_{i,j}$: The amount of supplier j's internal resources required to produce one unit of raw material i,

q_j : Total amount of resources reserved by supplier j from manufacturer,

t_k : The amount of the manufacturer's internal resources required to produce one unit of product k,

Q: The total amount of manufacturers' resources,

$c_{i,j,l}$: Supplier j's unit price needed to provide one unit of raw material i within the order interval $[d^S_{i,j,l}, d^H_{i,j,l}]$. Price level l=1,2, ..., L,

g_j : Manufacturer's management costs when using supplier j,

$k_{i,j}$: Fixed order setup costs for raw material i from supplier j,

$h_{i,j}$: Periodic holding costs per unit associated with raw material i from supplier j,

y_k : The number of units of product k to be produced in the period,

$x_{i,j,l}$: The number of units of raw material i purchased from supplier j at the price level l,

$u_{i,j,l}$: 1 if the manufacturer buys any amount of raw material i from supplier j at price level l,

$u_{i,j}$: 1 if raw material i is purchased from supplier j; 0 otherwise,

v_j : 1 if the manufacturer buys any raw materials from supplier j, 0 otherwise,

$w_{i,j}$: Order quantity of raw material i from supplier j at each cycle,

z_k : Demand of final product k in the next period of horizon.

3.1.3.3 Formulation

Objective function:

$$Re = \sum_{k=1}^{K}\left\{\int_0^{y_k}[r_k z_k - b_k(y_k - z_k)]f(z_k)dz_k + \int_{y_k}^{\infty}[r_k y_k - a_k(z_k - y_k)]f(z_k)dz_k\right\},$$

$$Maximize\ Re$$

$$-\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{l=1}^{L} c_{i,j,l}x_{i,j,l} - \sum_{k=1}^{K} e_k y_k - \sum_{j=1}^{J} g_j v_j$$

$$-\sum_{i=1}^{I}\sum_{j=1}^{J}\frac{k_{i,j}\sum_{l=1}^{L} x_{i,j,l}}{w_{i,j}} - \sum_{i=1}^{I}\sum_{j=1}^{J}\frac{h_{i,j}w_{i,j}\sum_{l=1}^{L} x_{i,j,l}}{2\sum_{k=1}^{K} m_{i,k}y_k}$$

Where Re is the manufacturer's revenue, the second item is the cost of purchasing the raw material. The third and the fourth items are the production and management costs associate with the suppliers, respectively. While the last two items are setup costs and inventory costs respectively.

Capacity Limits Constraints:

$$\sum_{k=1}^{K} m_{i,k} y_k \le \sum_{j=1}^{J} \sum_{l=1}^{L} x_{i,j,l} \ , i = 1, \dots, I,$$

$$\sum_{i=1}^{I} n_{i,j} \sum_{l=1}^{L} x_{i,j,l} \le q_j v_j \ , \ j = 1, \dots, J,$$

$$\sum_{k=1}^{K} t_k y_k \le Q \ ,$$

$$x_{i,j,l} \ , y_k \ge 0;$$

The first constraint ensures that there are enough raw materials available for production. The second ensures the capacity of each supplier will not be exceeded. The third represents the manufacturer's capacity constraint.

Quality Discount Constraints:

$$x_{i,j,l} \le d_{i,j,l}^{H} u_{i,j,l} \ , \forall \ i, j, l$$

$$x_{i,j,l} \ge d_{i,j,l}^{S} u_{i,j,l} \ , \forall \ i, j, l$$

$$\sum_{i=1}^{L} u_{i,j,l} = u_{i,j}, \forall \ i, j$$

$$v_j \ge u_{i,j} \ , \forall \ i, j$$

$$w_{i,j} \leq \sum_{l=0}^{L} x_{i,j,l} \text{ },\forall\, i,j,l$$

$$u_{i,j,l}\text{ },u_{i,j},v_i \in \{0,1\}\text{ },\forall\, i,j,l$$

The first two constraints ensure the price level for the amount purchased from the supplier within the discount interval offered. The third ensures that only one discount level will be applied to one offer, and the fourth indicates the internal logic relation between two binary variables. The last constraint guarantees that the order quantity at each interval will not exceed the total amount purchased from the supplier during the period.

3.1.3.4 Solution Background

Zhang (2007) developed an iterative solution procedure by finding the optimal order quantity $w_{i,j}$ after relaxing the order quantity constraint:

$$w_{i,j} \leq \sum_{l=0}^{L} x_{i,j,l} \text{ },\forall\, i,j,l$$

The optimal order quantity becomes:

$$w_{i,j}^{*} = \sqrt{\frac{2k_{i,j}\sum_{k=1}^{K} m_{i,k}y_k}{h_{ij}}}$$

And the inventory cost and order set up costs can be expressed as follows:

$$Inventory\ and\ setup\ cost\text{: } \sum_{i}\sum_{j}\left(\sqrt{\frac{2k_{ij}h_{ij}}{\sum_k m_{ik}y_k}} * \sum_{l} x_{ijl}\right)$$

By doing this, variable $w_{i,j}$ is eliminated from the problem, which alleviates computational effort. Detail of the iterative algorithm can be found in Zhang (2007). It is worth mentioning that this research is based on the iterative algorithm. The main effort of the algorithm lies in the computation of the inner Mixed Integer Nonlinear Problem (MINLP) model. Zhang (2007) solved the model using GAMS modeling language and its MINLP solvers---SBB.

Due to the stochastic factors, the uncertain demand, and the unavailability of the internal functions in GAMS, external functions were developed to calculate the integration functions in the objective function, or simply Re. However, there are still problems remaining with the GAMS commercial solver for this special MINLP. The problem is extremely sensitive when changing of parameters due to the external integration function. As GAMS-SBB is commercial software, it is difficult to deal with the solver itself.

3.2 Foundation of Open Source Solution Approaches

The open source solution approach is based on the Open Source MINLP package --- BONMIN which acts as the solver, and AMPL, which appears as the modeling platform. Refer to Chapter 2 for a BONMIN discussion; the preceding discussion will focus on the implementation process of the problem using AMPL and BONMIN.

The open source solution approach to the stochastic supply chain problem (Zhang, 2007) modeled as MINLP is realized through two methodologies,

- External function based open-source approach
- Triangular approximation based open-source approach

These two approaches are illustrated separately in detail in the following sections. In this research, the software is operated under Linux x86_64 GNU/Linux, which is accessed through SSH (Secure Shell).

3.2.1 AMPL

Developed by Bell Laboratories, AMPL (**A** **M**athematical **P**rogramming **L**anguage) is a comprehensive and powerful algebraic modeling language for linear and nonlinear optimization problems for discrete or continuous variables (www.ampl.com). AMPL's flexibility and convenience make it ideal for rapid prototyping and model development, its speed and control options make it an efficient choice for repeated production runs (www.ampl.com). So far, the 64 bit AMPL student version is obtained, which can accommodate less than 300 hundred variables.

The user friendly modeling language makes it easy to express complicated mathematical problem, as it resembles natural language. AMPL has the following basic files to model problems.

In general, like most popular optimization modeling systems, AMPL supports the most basic model types such as Linear Programming, Mixed Integer Programming, Constrained and Unconstrained Nonlinear Programming, Mixed Integer Nonlinear Programming and Quadratic Programming. Moreover, AMPL can be operated under either Microsoft Windows System or Unix-like systems. In this research, the latter is adopted. In order to model a problem in AMPL, one should know the following three types of files as seen in Table 1:

.mod file	The file to define parameters, variables, objective functions and constraints

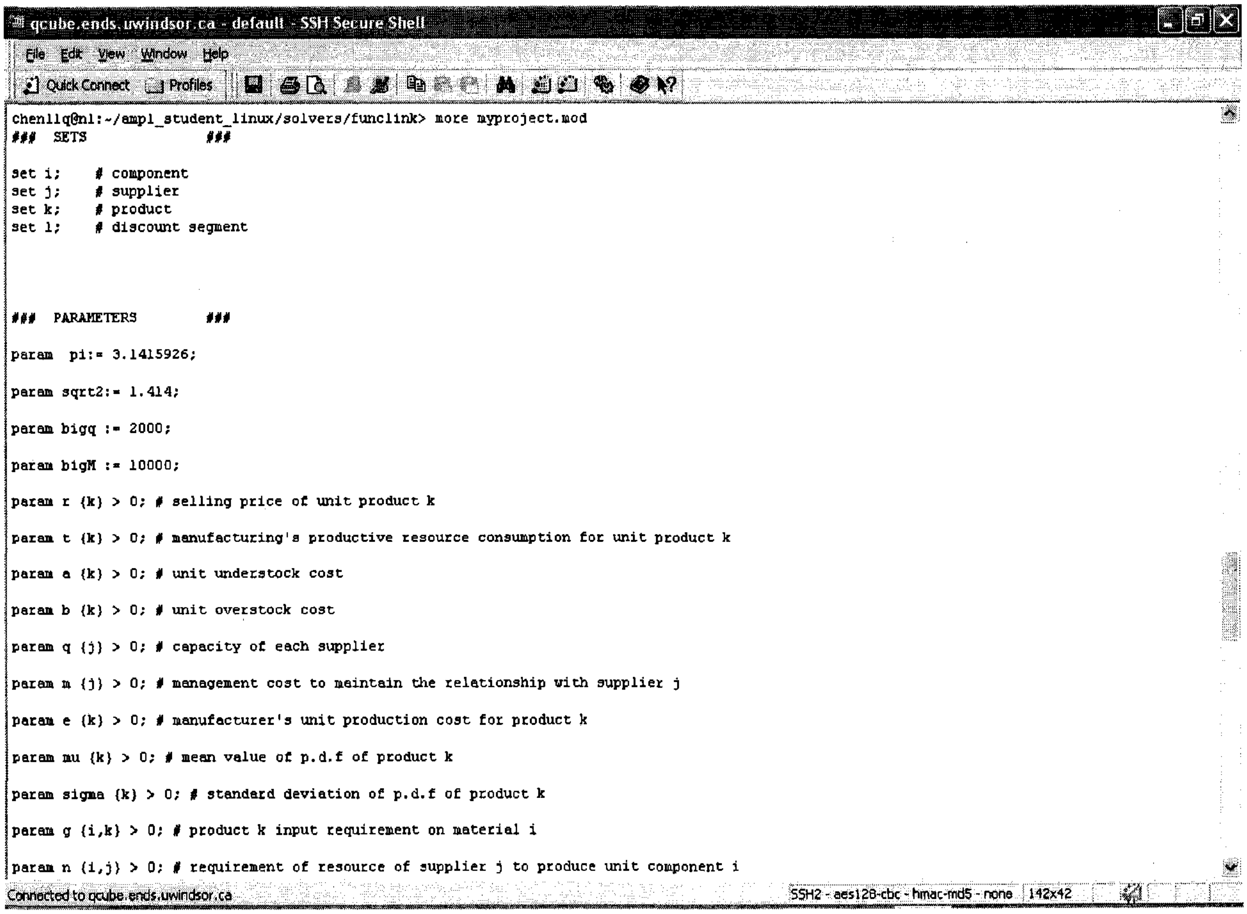
.dat file	The file to declare the values for each parameter
.x file	The file incorporating a group of command to implement and run the model

Table 1: Main AMPL files

In .mod file, the following inputs will be needed to define an AMPL model (Fourer et al. 2003):

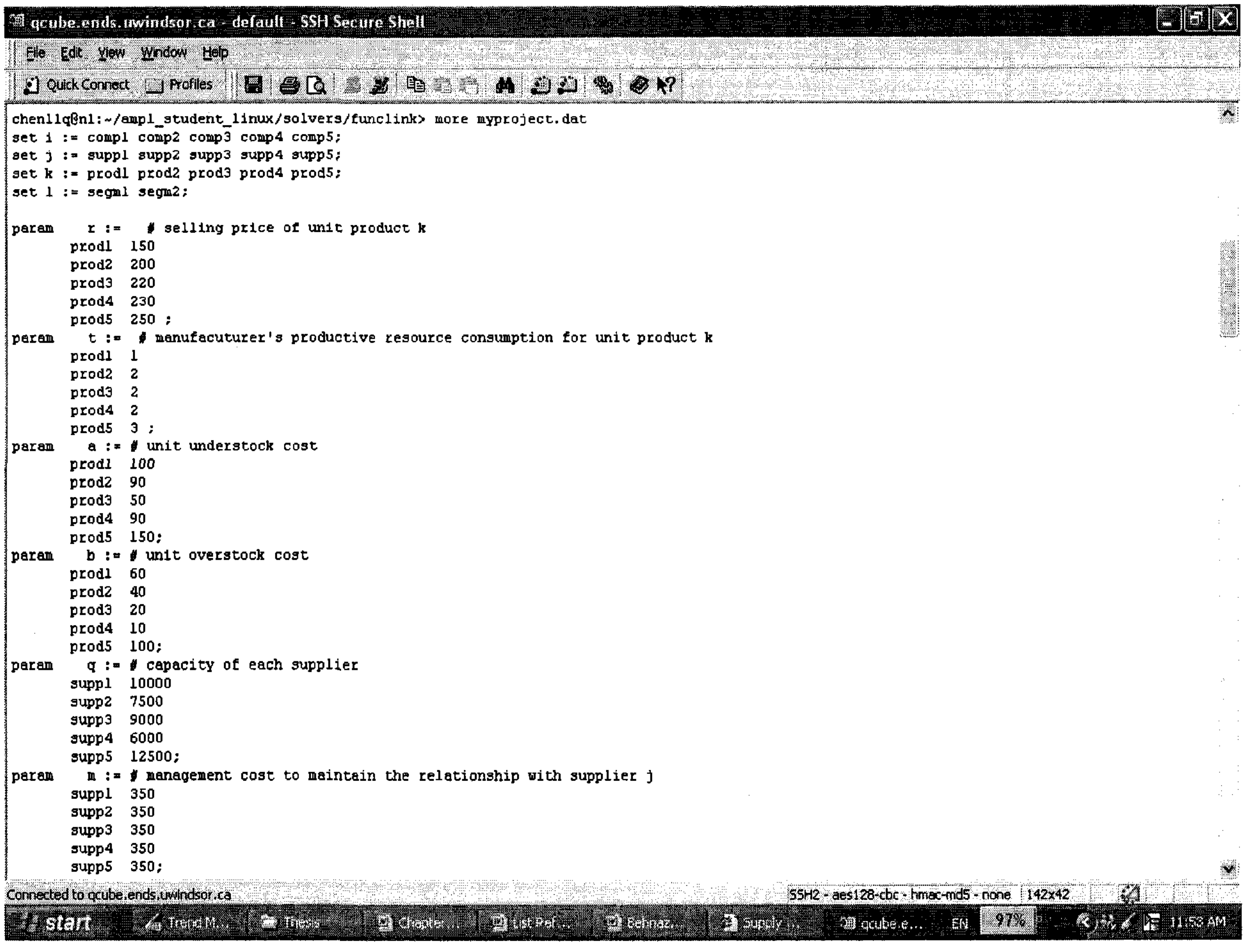
- *set* : Declares a set name;
- *param* : Declares a parameter which may be a single scalar value such as or a collection of values indexed by a set;
- *var* : Names a collection of variables;
- *maximize* or *minimize* followed by the name of objective function, which is arbitrary as long as it is not violating the syntax;
- *Subject to* followed by the name of the constraint.

Figure 2 is a snapshot of part of .mod model to illustrate the environment. All values of parameters must be defined in the .dat file. However, it does not mean that one has to have both a .mod file and a .dat file in order to build a model successfully, the separate model file and data file makes it easier to modify the model or data in the future, hence it is recommended, Figure 3 demonstrates of a .dat file.


qcube.ends.uwindsor.ca - default - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles
Connected to qcube.ends.uwindsor.ca
SSH2 - aes128-cbc - hmac-md5 - none 142x42

```
chenllq@nl:~/ampl_student_linux/solvers/funclink> more myproject.mod
###  SETS            ###

set i;    # component
set j;    # supplier
set k;    # product
set l;    # discount segment



###  PARAMETERS        ###

param  pi:= 3.1415926;

param sqrt2:= 1.414;

param bigq := 2000;

param bigM := 10000;

param r {k} > 0; # selling price of unit product k

param t {k} > 0; # manufacturing's productive resource consumption for unit product k

param a {k} > 0; # unit understock cost

param b {k} > 0; # unit overstock cost

param q {j} > 0; # capacity of each supplier

param m {j} > 0; # management cost to maintain the relationship with supplier j

param e {k} > 0; # manufacturer's unit production cost for product k

param mu {k} > 0; # mean value of p.d.f of product k

param sigma {k} > 0; # standard deviation of p.d.f of product k

param g {i,k} > 0; # product k input requirement on material i

param n {i,j} > 0; # requirement of resource of supplier j to produce unit component i
```

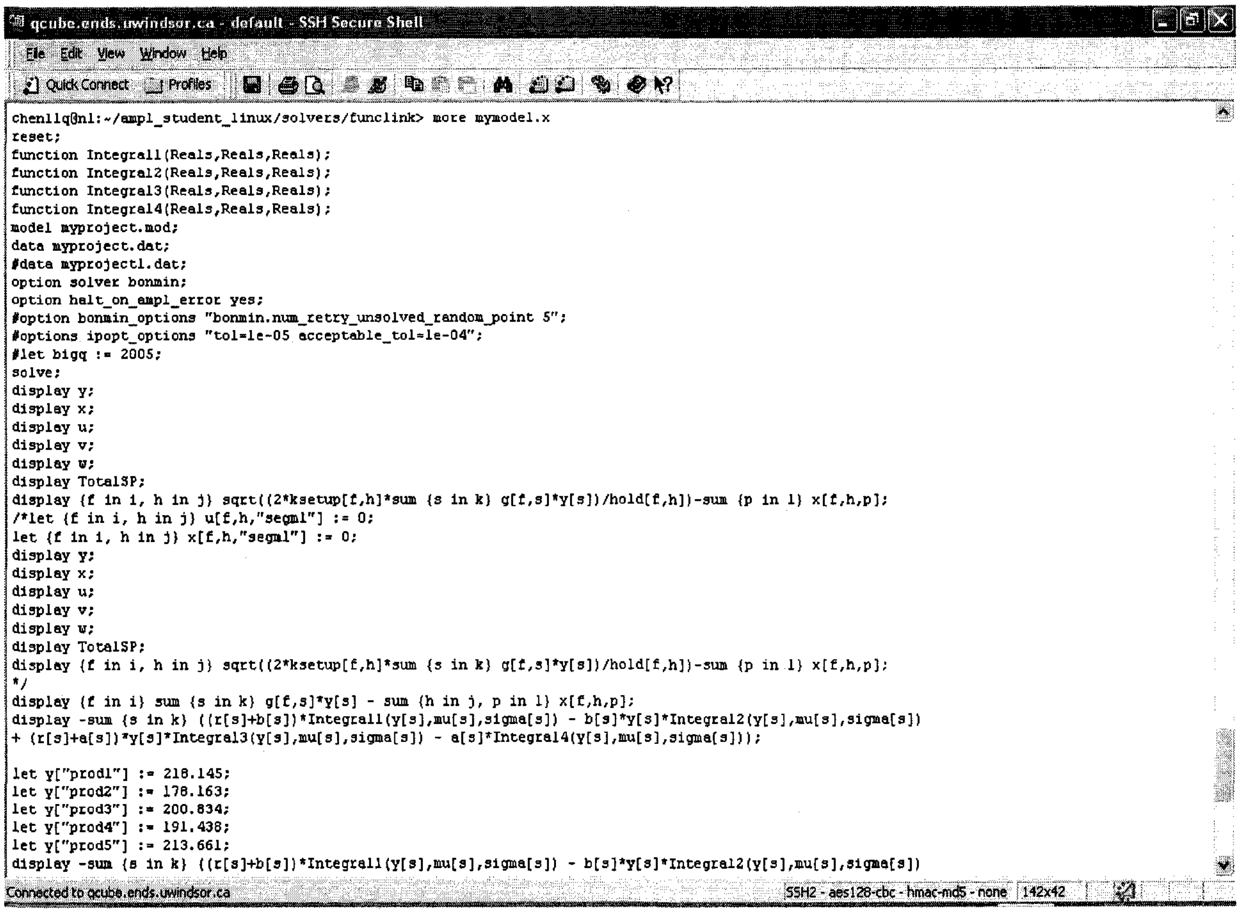
Figure 2: AMPL .mod file illustration (SSH snapshot)


qcube.ends.uwindsor.ca - default - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles
Connected to qcube.ends.uwindsor.ca
SSH2 - aes128-cbc - hmac-md5 - none
142x42
start
Trend M...
Thesis
Chapter...
List Ref...
Behnaz...
Supply...
qcube.e...
EN
97%
11:53 AM

```
chen11q@n1:~/ampl_student_linux/solvers/funclink> more myproject.dat
set i := comp1 comp2 comp3 comp4 comp5;
set j := supp1 supp2 supp3 supp4 supp5;
set k := prod1 prod2 prod3 prod4 prod5;
set l := segm1 segm2;

param    r :=   # selling price of unit product k
       prod1  150
       prod2  200
       prod3  220
       prod4  230
       prod5  250 ;
param    t :=  # manufacuturer's productive resource consumption for unit product k
       prod1  1
       prod2  2
       prod3  2
       prod4  2
       prod5  3 ;
param    a := # unit understock cost
       prod1  100
       prod2  90
       prod3  50
       prod4  90
       prod5  150;
param    b := # unit overstock cost
       prod1  60
       prod2  40
       prod3  20
       prod4  10
       prod5  100;
param    q := # capacity of each supplier
       supp1  10000
       supp2  7500
       supp3  9000
       supp4  6000
       supp5  12500;
param    m := # management cost to maintain the relationship with supplier j
       supp1  350
       supp2  350
       supp3  350
       supp4  350
       supp5  350;
```

Figure 3: AMPL .dat file illustration (SSH Snapshot)

Under the Unix-like system, AMPL is accessed through command lines which are both tedious and time consuming. In order to prevent the repetitive work, AMPL users are allowed to create .x files as a pool of commands to use to implement the model and still make modifications to either model or data files. In addition, different solver options may be declared and modified in .x files. Figure 4 provides you an example of a typical .x file.


qcube.ends.uwindsor.ca - default - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles
Connected to qcube.ends.uwindsor.ca
SSH2 - aes128-cbc - hmac-md5 - none 142x42

```
chenllq@nl:~/ampl_student_linux/solvers/funclink> more mymodel.x
reset;
function Integrall(Reals,Reals,Reals);
function Integral2(Reals,Reals,Reals);
function Integral3(Reals,Reals,Reals);
function Integral4(Reals,Reals,Reals);
model myproject.mod;
data myproject.dat;
#data myprojectl.dat;
option solver bonmin;
option halt_on_ampl_error yes;
#option bonmin_options "bonmin.num_retry_unsolved_random_point 5";
#options ipopt_options "tol=le-05 acceptable_tol=le-04";
#let bigq := 2005;
solve;
display y;
display x;
display u;
display v;
display w;
display TotalSP;
display {f in i, h in j} sqrt((2*ksetup[f,h]*sum {s in k} g[f,s]*y[s])/hold[f,h])-sum {p in l} x[f,h,p];
/*let {f in i, h in j} u[f,h,"segml"] := 0;
let {f in i, h in j} x[f,h,"segml"] := 0;
display y;
display x;
display u;
display v;
display w;
display TotalSP;
display {f in i, h in j} sqrt((2*ksetup[f,h]*sum {s in k} g[f,s]*y[s])/hold[f,h])-sum {p in l} x[f,h,p];
*/
display {f in i} sum {s in k} g[f,s]*y[s] - sum {h in j, p in l} x[f,h,p];
display -sum {s in k} ((r[s]+b[s])*Integrall(y[s],mu[s],sigma[s]) - b[s]*y[s]*Integral2(y[s],mu[s],sigma[s])
+ (r[s]+a[s])*y[s]*Integral3(y[s],mu[s],sigma[s]) - a[s]*Integral4(y[s],mu[s],sigma[s]));

let y["prodl"] := 218.145;
let y["prod2"] := 178.163;
let y["prod3"] := 200.834;
let y["prod4"] := 191.438;
let y["prod5"] := 213.661;
display -sum {s in k} ((r[s]+b[s])*Integrall(y[s],mu[s],sigma[s]) - b[s]*y[s]*Integral2(y[s],mu[s],sigma[s])
```

Figure 4: AMPL .x file illustration (SSH Snapshot)

3.2.2 BONMIN

Another component shared by the two aforementioned approaches is BONMIN, which is the MINLP solver for the problem. Since BONMIN has been introduced briefly within the literature review, the implementation of BONMIN is focused.

Download and installation procedures under various systems can be found in the BONMIN users manual (Bonami and Lee, 2006). It can be done through the following ways:

- From a command line through .nl file (Gay, 2005)
- From Modeling Language AMPL

- By invoking it from C/C++ programming

In this research, we use AMPL modeling language to interact with BONMIN because of the user-friendliness features of AMPL.

To use BONMIN from AMPL, the directory where the BONMIN executable is located has to be added in your environment variable $PATH to issue the command:

option solver bonmin;

in the AMPL environment. Then BONMIN will be used to solve the model loaded in AMPL. After the optimization process is finished, the values of the variables in the best-known or optimal solution can be accessed in AMPL. If the optimization is interrupted with <CTRL-C>, then the best known solution is accessible (Bonami and Lee, 2006). Different parameters of BONMIN can also be changed through AMPL commands. What is also worth mentioning is that BONMIN comes with a parameter setting file called Bonmin.opt, which should be put under the same directory as .mod, .dat and/or .x files.

3.3 External Function Based Open Source Approach

For the model in (Zhang, 2007), the integration function calculation becomes the key point in the optimization process, which is unavailable among the AMPL internal functions. Therefore, external functions must be created along with AMPL to model the problem with BONMIN as the open source solver. The structure for solving the MINLP model is illustrated in Figure 5.

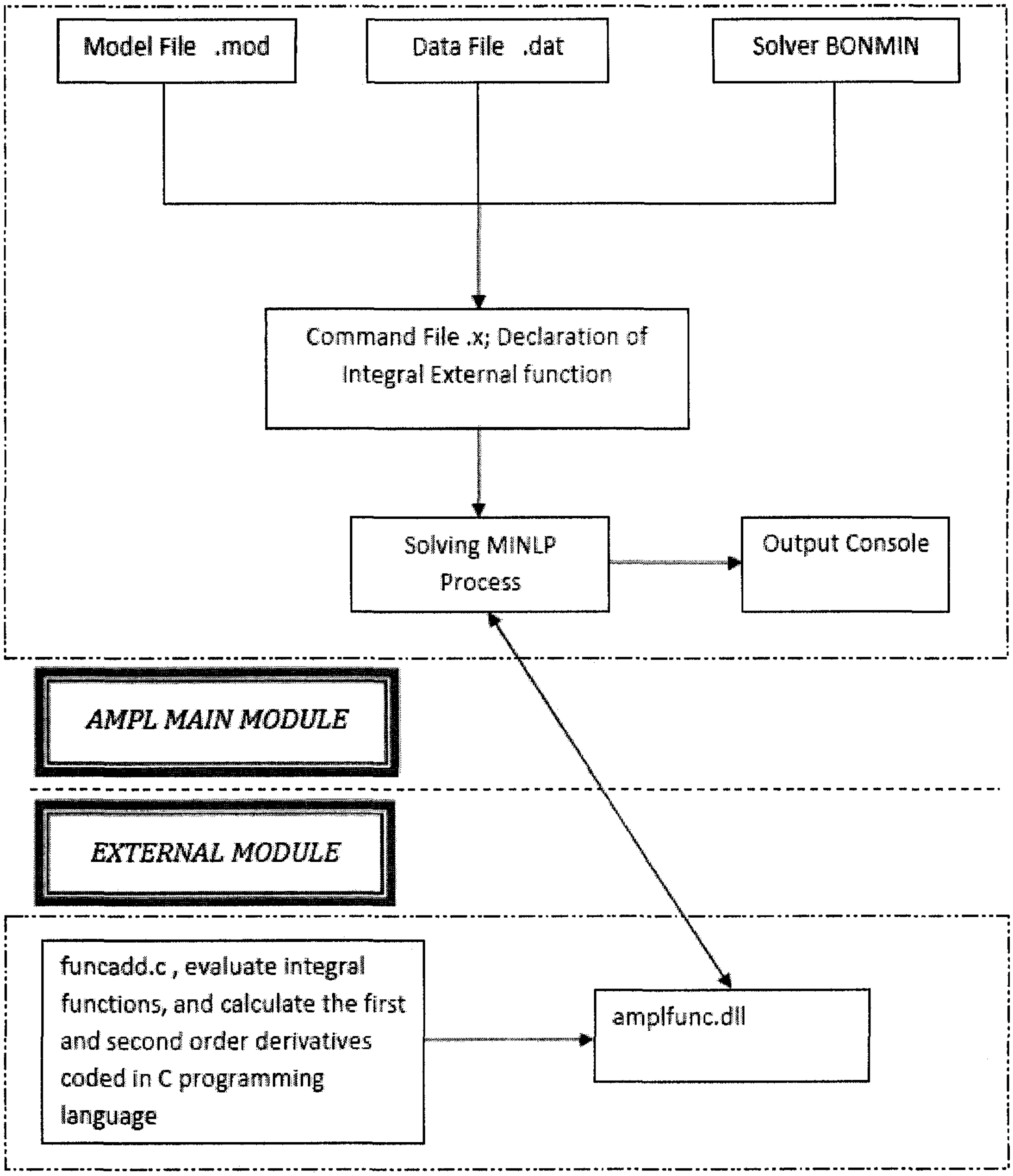

Model File .mod
Data File .dat
Solver BONMIN
Command File .x; Declaration of Integral External function
Solving MINLP Process
Output Console
AMPL MAIN MODULE
EXTERNAL MODULE
funcadd.c , evaluate integral functions, and calculate the first and second order derivatives coded in C programming language
amplfunc.dll

Figure 5: External function based open source BONMIN approach in AMPL

The C code is used to evaluate the external functions and the derivatives are derived from the Romberg Integration Algorithm, which was coded originally by Press et al. (1992). the code is tailored slightly to accommodate into the 4 integration functions that have 3 parameters in each of them. There are certain rules to make the external functions; please refer to the following link for more information (www.netlib.org/ampl/solvers/funclink/funcadd.c). In order to use BONMIN within AMPL, AMPL Solver Library has to be downloaded, which can be accessed from www.netlib.org. The same thing can also be done by completing the compiling procedure of the BONMIN package according to the installation instruction (Bonami and Lee, 2006).

Four external integration functions are developed according to the objective function, they are as follows:

$$Integral1(y[k],\mu[k],\sigma[k]) = \int_0^{y[k]} \frac{z_k}{\sigma[k]\sqrt{2\pi}} e^{-\frac{(z_k-\mu[k])^2}{2\sigma[k]^2}} dz_k\,,$$

$$Integral2(y[k],\mu[k],\sigma[k]) = \int_0^{y[k]} \frac{1}{\sigma[k]\sqrt{2\pi}} e^{-\frac{(z_k-\mu[k])^2}{2\sigma[k]^2}} dz_k\,,$$

$$Integral3(y[k],\mu[k],\sigma[k]) = \int_{y[k]}^{+\infty} \frac{1}{\sigma[k]\sqrt{2\pi}} e^{-\frac{(z_k-\mu[k])^2}{2\sigma[k]^2}} dz_k\,,$$

$$Integral4(y[k],\mu[k],\sigma[k]) = \int_{y[k]}^{+\infty} \frac{z_k}{\sigma[k]\sqrt{2\pi}} e^{-\frac{(z_k-\mu[k])^2}{2\sigma[k]^2}} dz_k\,,$$

3.4 Triangular Approximation Open Source Approach

The model in Zhang (2007) could be considered as an extension to the multi-product, multi-constraint newsvendor models and most algorithm approaches for newsvendor problem are based on the characteristics of the relaxed constraint version and the Lagrange Multiplier Method for the nonlinear problem. Neither of these ideologies can be efficiently adapted to Zhang's problem, because the existence of the additional raw material variables. This is in addition to the product variables and another nonlinear item, which is inventory cost and transportation or setup cost. When there is no provided efficient algorithm approach to deal with integration functions in objective function, it is impossible to tackle larger scale problems. Although the approach in 3.3 provides a different open source approach which is different from commercial software approach such as GAMS in Zhang (2007), it still does not get independent of external functions, the existence of which is actually the source of vulnerability when it faces large scale problems, because it still takes a relatively longer time to solve a small size problem.

Apparently, better solution approaches are needed to to solve large scale problems. Areeratchakul and Abedel-Malek (2007) developed a triangular approximation approach to a multiproduct, multi-constraint newsvendor problem. The author used an approximation of the integration of the newsvendor problem, which is similar to the model in Zhang (2007). By doing this, the objective function becomes a quadratic problem with another single nonlinear part, which is a combined item of inventory cost and setup cost based on the algorithm in Zhang (2007). The following is the reasoning process of the triangular approximation approaches.

In the formulation of the problem, which is based on Areeratchakul and Abedel-Malek (2006), the following is defined:

$f(z_k)$ is the density function of the demand distribution of product k;

$F(y_k) = \int_0^{y_k} f(z_k)dz_k$, which is defined as a cumulative distribution function.

$$Re = \sum_{k=1}^{K}\left\{\int_0^{y_k}[r_k z_k - b_k(y_k - z_k)]f(z_k)dz_k + \int_{y_k}^{\infty}[r_k y_k - a_k(z_k - y_k)]f(z_k)dz_k\right\},$$

$$Re_k = (r_k + b_k)\int_0^{y_k} z_k f(z_k)dz_k - b_k y_k \int_0^{y_k} f(z_k)dz_k + (a_k + r_k)y_k \int_{y_k}^{\infty} f(z_k)dz_k - a_k \int_{y_k}^{\infty} z_k f(z_k)\, dz_k$$

$$= (r_k + b_k)\int_0^{y_k} z_k f(z_k)dz_k - b_k y_k F(y_k) + (a_k + r_k)y_k[1 - F(y_k)] - a_k\left(E(z_k) - \int_0^{y_k} z_k f(z_k)dz_k\right)$$

$$= (a_k + b_k + r_k)\int_0^{y_k} z_k f(z_k)dz_k - (a_k + b_k + r_k)y_k F(y_k) + (a_k + r_k)y_k - a_k E(z_k) \qquad (1)$$

Integrating the function by parts, we are able to obtain

$$\int_0^{y_k} z_k f(z_k)dz_k = y_k F(y_k) - \int_0^{y_k} F(z_k)dz_k \qquad (2)$$

Substituting equation (2) into (1), we get:

$$Re_k = (a_k + r_k)y_k - a_k E(z_k) - (a_k + b_k + r_k)\int_0^{y_k} F(z_k)dz_k$$

Correspondingly, we have

$$Re = \sum_{k=1}^{K}\left\{(a_k + r_k)y_k - a_k E(z_k) - (a_k + b_k + r_k)\int_0^{y_k} F(z_k)dz_k\right\} \qquad (3)$$

Now we will make the approximation of $\int_0^{y_k} F(z_k)dz_k$ using the triangular approach:

$$\int_0^{y_k} F(z_k)dz_k \approx \frac{1}{2}\left(y_k - y_{l,k}\right)\left[\Delta_k\left(y_k - y_{l,k}\right)\right] \qquad (4)$$

Therefore, we have the error function:

$$error = \int_0^{y_k} F(z_k)dz_k - \frac{1}{2}(y_k - y_{l,k})[\Delta_k(y_k - y_{l,k})] \tag{5}$$

Then Re could be expressed as the following way:

$$Re = \sum_k^K (A_k y_k^2 + B_k y_k + C_k) \tag{6}$$

And the problem becomes how to determine the different parameters: A_k, B_k, C_k.

Substituting (3) into (6) we have:

$$A_k = -\frac{1}{2}\Delta_k(a_k + b_k + r_k) \tag{7}$$

$$B_k = (a_k + b_k + r_k)\Delta_k y_{l,k} + a_k + r_k \tag{8}$$

$$C_k = -\left(\frac{a_k+b_k+r_k}{2}\right)\Delta_k y_{l,k}^2 - a_k E(z_k) \tag{9}$$

3.4.1 Taylor Series based Triangular Approximation Approach:

Getting the triangular approach through the Taylor Series of $F(z_k)$ at μ_k, it is assumed that the majority of z_k distributes around μ_k. Please see Figure 6 as illustration of this idea, where the shaded area represents the triangular approximation of $\int_0^{y_k} F(z_k)dz_k$.

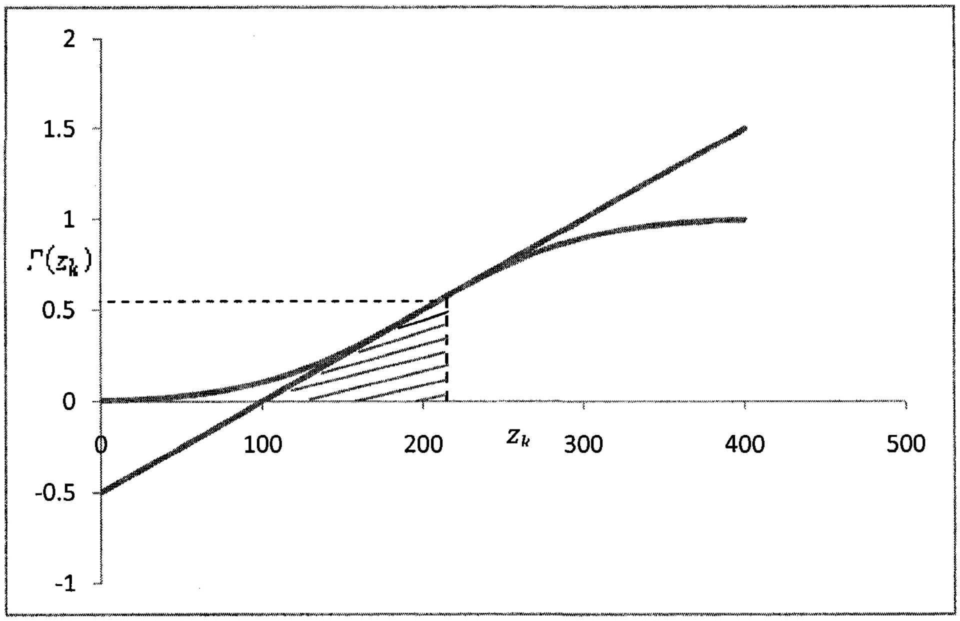


Figure 6: Taylor series based triangular approximation approach

The function of straight line in the Figure 6 can be expressed as:

$$F(z_K) - F(\mu_k) = f(\mu_k)(z_k - \mu_k); \tag{10}$$

Let $F(z_k) = 0$, then

$$y_{l,k} = \mu_k - \frac{F(\mu_k)}{f(\mu_k)}, \text{ if } \mu_k - \frac{F(\mu_k)}{f(\mu_k)} \geq 0; \text{ Otherwise, } y_{l,k} = 0 \tag{11}$$

$$\Delta_k = f(\mu_k) \tag{12}$$

Substituting (11) (12) into (7) (8) (9), the followings for the normal distribution function are obtained:

$$A_k = -\frac{1}{2} f(\mu_k)(a_k + b_k + r_k) \tag{13}$$

$$B_k = (a_k + b_k + r_k)[\mu_k f(\mu_k) - F(\mu_k)] + a_k + r_k \tag{14}$$

$$C_k = -\frac{a_k + b_k + r_k}{2} \times \frac{[\mu_k f(\mu_k) - F(\mu_k)]^2}{f(\mu_k)} - a_k E(z_k) \tag{15}$$

Apparently, $E(z_k) = \mu_k$, therefore, given the specific parameters of product k and the corresponding distribution function, the quadratic approximation of Re can be obtained.

3.4.2 Covering Range Based Triangular Approximation Approach:

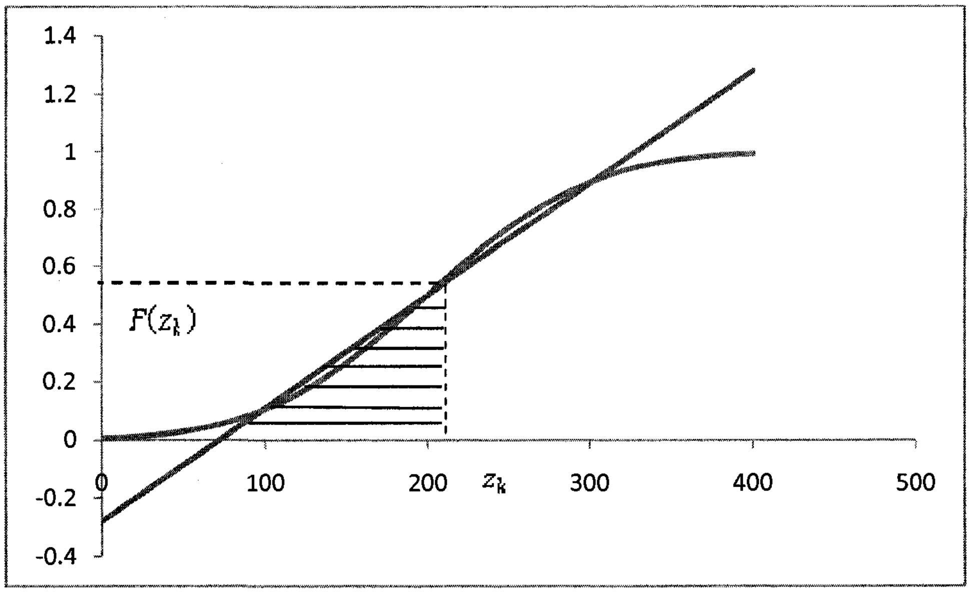


Figure 7: Covering range based triangular approximation approach

The following formula is used to calculate Δ_k, and $y_{l,k}$ as demonstrated in Figure 7, where the shaded area stands for the triangular approximation of $\int_0^{y_k} F(z_k)dz_k$,

$\Delta_k = \frac{0.9-0.1}{F^{-1}(0.9)-F^{-1}(0.1)}$ and

$y_{l,k} = F^{-1}(0.9) - \frac{0.9}{\Delta_k}$, if $y_{l,k} \geq 0$; 0, otherwise.

Based either of the two approximation approaches, the integration functions can be transferred into an equivalent quadratic function with respect to y_k.

4. RESULTS AND ANALYSIS TO BONMIN BASED APPROACHES

4.1 Results and Comparison of Three Proposed Approaches

In this chapter, it is illustrated how the three approaches perform compared to the approach used in Zhang (2007), and these three approaches are also validated through the model and results in Kim et al. (2002). In order to compare these three approaches, all models are run under exactly the same BONMIN options settings. The results of the two BONMIN based approaches under different scenarios are reported and compared with AMPL external function based approach. The specifications of the example problems can be found in Appendix tables.

The notation "EFO", "TSTO", "CRTO" represents "External Function based Open Source Approach"; "Taylor Series based Triangular Approximation Open source Approach" and "Covering Range based Triangular Approximation Open source Approach" respectively.

Among the computational tests, the Branch-and-Bound algorithm option of BONMIN is selected as the algorithm among four available options. From table 2, we can easily see the difference among the three approaches. Please note that the solution values of the rest of the $x_{i,j,l}$ variables which are not listed are zero, and the values of the binary variables become obvious according to their corresponding $x_{i,j,l}$ values.

Since the problem is a convex problem, BONMI-MINLP algorithm guarantees the global optimum for it. Given that the correct evaluation and derivatives information is provided, the solution, which is obtained by EFO approach, is the global optimal solution. However, the time and the number of iterations EFO takes are much more than those of TSTO and

CRTO. This can predict EFO's unreliability when it comes to larger scale problems. TSTO approach provides almost exactly the same solution for y_s and $x_{i,j,l}$ as EFO, nevertheless, the objective function value deviates more from that of EFO compared with CRTO. Due to the following fact:

- TSTO and EFO shares the same formulation except for the different ways of expressions of the integral part of objective function;
- Almost exactly the same values of optimal solutions for all variables;

BB algorithm	Approaches (The manufacturer capability is set to be 2000)			
Criteria	EFO	TSTO	CRTO	GAMS
y_1	216.972	216.923	218.145	217.572
y_2	177.939	178.054	178.163	177.883
y_3	201.189	201.403	200.834	201.134
y_4	189.112	188.439	191.438	189.229
y_5	215.516	215.762	213.661	215.312
$x_{1,4,2}$	2051.11	2051.83	2049.38	2051.595
$x_{2,5,2}$	1773.31	1773.85	1773.06	1773.341
$x_{3,1,2}$	2179.95	2177.8	2191.01	2181.842
$x_{4,2,2}$	2443.66	2443.07	2445.08	2444.230
$x_{5,3,2}$	2177.94	2178.05	2178.16	2177.883
O. B. Value	62600.9	72832.1	62445.7	67670.419
Number of Nodes Visited	15	15	15	N/A

CPU Time	21.81	0.37	0.37	N/A
Number of Iterations	866	274	285	N/A

Table 2: Comparison among different approaches when Q=2000

It can be concluded that the difference of final objective function values between TSTO and EFO derives from the triangular approximation of integral functions. However, TSTO delivers much better performance regarding computing time and the number of iterations than EFO. Similar phenomenon can be observed for CRTO with respect to computing time and number of iterations. What is worth pointing out is that CRTO delivers more accurate objective function value than TSTO. Same numbers of nodes being observed can be simply explained by the fact that the same branching and node selecting rules are used for all three approaches during the solving process. GAMS solutions to the same problem are obtained from Zhang (2007), by comparing the results of variables, one can easily see that the open-source approach delivers as good solutions as GAMS.

BB algorithm	Approaches (The manufacturer capability is set to be 2050)		
Criteria	EFO	TSTO	CRTO
y_1	219.516	219.08	219.967
y_2	181.616	181.085	184.118
y_3	206.083	205.427	207.235
y_4	193.096	191.472	195.127
y_5	220.228	219.762	219.025
$x_{1,4,2}$	2092.68	2086.28	2097.96

$x_{2,5,2}$	1810.08	1804.18	1819.97
$x_{3,1,2}$	2220.47	2210.48	2234.9
$x_{4,2,2}$	2493.01	2483.56	2500
$x_{5,3,2}$	2223.41	2215.42	2234.12
Objective Function Value	62680.1	72889	62724
Number of Nodes Visited	16	15	18
CPU Time	2215.70	0.34	0.4
Number of Iterations	4284	266	320

Table 3: Comparison among different approaches when Q=2050

Similar solution can be observed in Table 3 when the capacity of manufacturing resources is changed to 2050 per year; however, the time it takes for EFO is 6000 times longer than those of TSTO and CRTO. This observation indicates that EFO is not suitable for larger scale problems and very sensitive to the parameters changes, even for smaller scale problems.

Since the exact evaluation of objective function is available through EFO, it is possible to combine TSTO and CRTO with EFO together to deliver both good solutions and more accurate objective values. Therefore, the combined approaches E-TSTO and E-CRTO are created. The results of implementation of this idea can be found in Table 4 and Table 5.

BB algorithm	Approaches (The manufacturer capability is set to be 2000)		
Criterion	EFO	TSTO	CRTO
Objective Function Value	62600.9	72832.1	62445.7
Criterion	EFO	E-TSTO	E-CRTO

Revised Objective	62600.9	62600.3	62590.2

Table 4: Comparison between combined approaches and original ones when Q=2000

BB algorithm	Approaches (The manufacturer capability is set to be 2050)		
Criteria	EFO	TSTO	CRTO
Objective Function Value	62680.1	72889	62724
Criteria	EFO	E-TSTO	E-CRTO
Revised Objective	62680.1	62676.6	62667

Table 5: Comparison between combined approaches and original ones when Q=2050

The results in Table 4 and Table 5 indicate that E-TSTO and E-CRTO provide higher quality solution in terms of both objective values and computing efforts than TSTO and CRTO.

In brief, table 6 reports the objective function values as the manufacturing capacity changes.

Q	EFO	E-TSTO	E-CRTO
1700	57205	57205	57188.7
1800	59958.9	59958.9	59942
1900	61754.2	61754.1	61739.4
2000	62600.9	62600.3	62590.2
2100	62680.1	62676.6	62660.2
2200	62680.1	62676.6	62660.2
2300	62680.1	62676.6	62660.2

2400	62680.1	62676.6	62660.2

Table 6: Objective function values for different approaches when Q changes

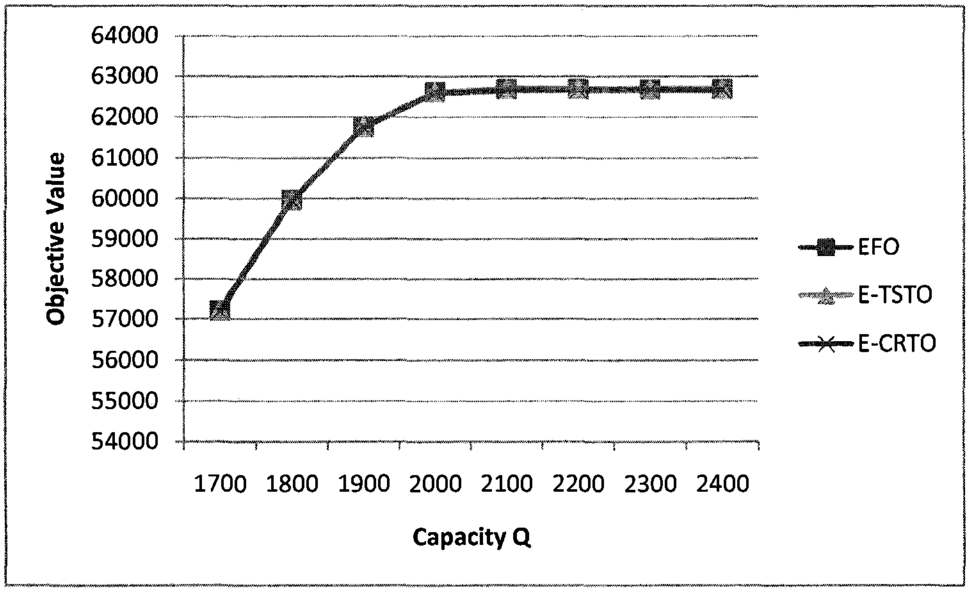


Figure 8: Graphical illustration of table 6

Figure 8 is drawn from the data in table 6, it can be noticed that E-TSTO and E-CRTO provides fairly good approximations as very close figure has also been reported in Zhang (2007), where the profit will not increase until the capacity Q roughly reaches 2100. Beyond 2100, the manufacturer's capability will not be fully used due to the resources constraints of suppliers.

From the results reported above, it can be concluded that E-TSTO and E-CRTO can be projected as more efficient and reliable approaches for larger scale problems.

4.2 Validation of Triangular Approximation Approach

In order to validate the triangular approximation approach to the integral part of the problem, experimental tests are carried out to compare the results of E-CRTO and analytical global solutions to the same test problems in Kim et al. (2002) (Only E-CRTO is used to demonstrate the accuracy and efficiency of triangular approximation approach for the sake of simplification). Due to the comparatively straight forward structure in Kim et al (2002), the global optimal solutions obtained by a proposed iterative algorithm is guaranteed in the paper. The following results indicate that the Triangular Approximation approach indeed provides fairly well enough solution compared to the global optimum in Kim et al (2002). Table 7 and Figure 9 demonstrates the objective function values of case 1 in Kim's (see APPENDIX A for specifications of case 1) with different manufacturing capacity Q. Both E-CRTO and EFO delivers exactly the same quality of solutions compared with the global optimum of case 1 in Kim et al. (2002), however, EFO uses comparatively longer computing time. The details of computing time and results are shown in Table 7 and readers are referred to Kim et al. (2002) for the original graph to compare.

Q	E-CRTO	EFO	E-CRTO(TIME, SEC)	EFO(TIME, SEC)
2300	2983.64	2996.48	0.01	>2
2400	3130.74	3138.58	0.01	>2
2500	3275.84	3279.97	0.01	>2
2600	3418	3419.75	0.01	>2
2700	3555.91	3556.42	0.01	>2
2800	3683.11	3683.03	0.01	>2
2900	3791.46	3791.47	0.01	>2
3000	3888.51	3888.63	0.01	>5
3100	3970.36	3970.54	0.01	>5
3200	4005.44	4005.62	0.01	>5
3300	4005.44	4005.62	0.01	>5
3400	4005.44	4005.62	0.01	>5
3500	4005.44	4005.62	0.01	>5
3600	4005.44	4005.62	0.01	>5
3700	4005.44	4005.62	0.01	>5

Table 7: Validation of triangular approach on Kim's case 1 model

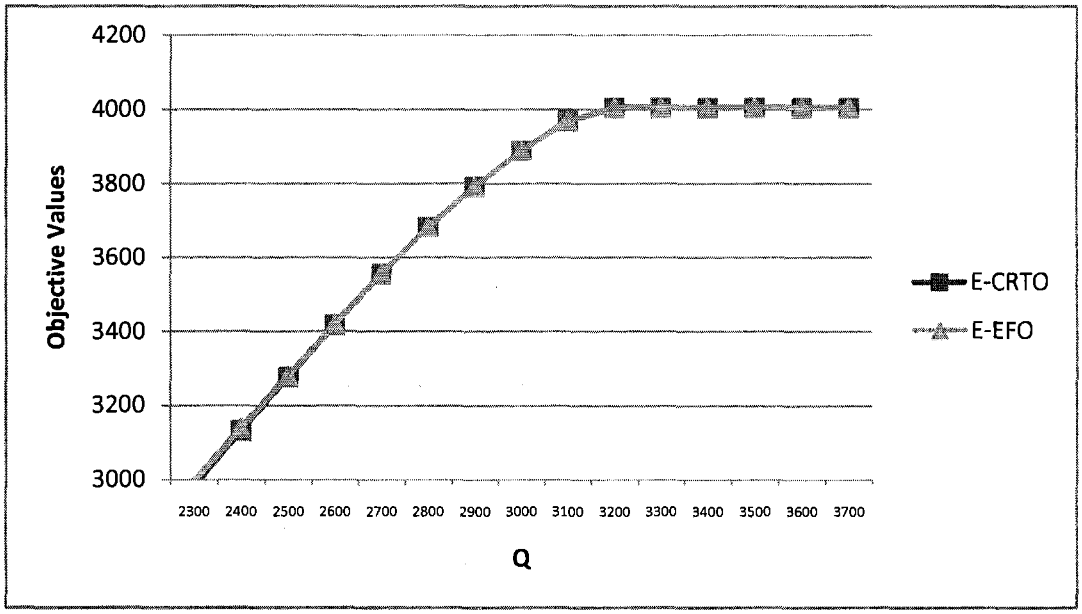


Figure 9: Graphical illustration of table 7

Case 2 in Kim et al.'s is also compared with E-CRTO to validate the triangular approximation approach. Please see table 30 in APPENDIX D for detailed results of production amount change among five different products along with change of manufacturing capacity Q. Parameters and problem definitions of Case 2 can be found in APPENDIX B. Figure 10 provides the same experiments as conducted in Kim et al.'s case two, very similar values and trends can also be observed for case two in Kim et al. (2002).

In this chapter, the proposed BONMIN based open-source approaches are validated by comparing the results in Kim et al. (2002) for both case 1 and case 2. In the future chapters, the BONMIN-CRTO approach will be used again as reference to test some larger scale problems.

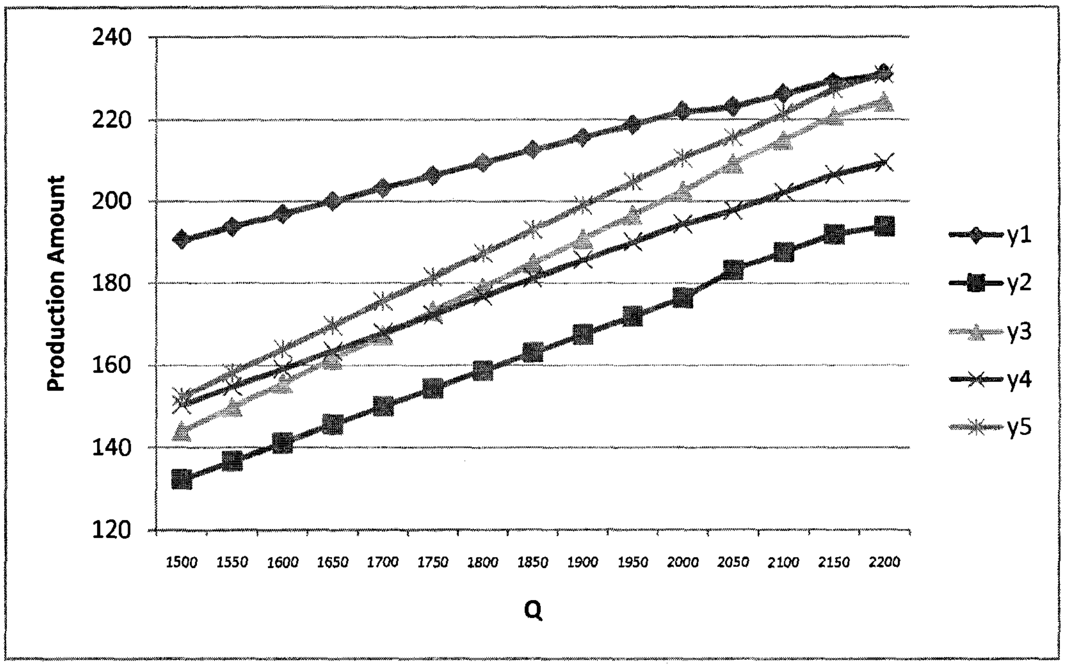


Figure 10: Graphical illustration of table 30

5. NONLINEAR BRANCH-AND-BOUND ALGORITHM

5.1 Motivation to the Development of B&B Algorithm

The approaches proposed in chapter 3 are closely correlated with open source MINLP package BONMIN. Although BONMIN complies with OSI, which makes it possible for every one to access the source code, it is still difficult for individual researchers to modify the open source MINLP code in BONMIN, especially, for the Branch-and-Bound code. Therefore, a solution approach which can successfully and independently control the solving procedure of the MINLP and take advantage of the specific feature of the model of Zhang (2007) itself becomes very interesting and important.

The currently available effective methods to solve MINLPs include Generalized Benders Decomposition (GDB), Outer Approximation (OA) and Branch-and-Bound (BB) (Kalvelagen, GAMS). Branch-and-Bound methods are used extensively for mixed-integer linear programming models and the basic method is directly applicable to MINLP. Similarly, the efforts that are needed to solve MINLP are good nonlinear solver, efficient integer branching and node selecting strategies. In order to improve the computing efficiency of the inner problem, triangular approximation is employed to transfer the integration function into a convex quadratic format, which is combined with the iterative algorithm to deal with the corresponding nonlinear part for inventory control. Therefore, the inner problem can be transferred into a convex quadratic problem since all the constraints are already linear. It is relatively easy to find the global optimal solution for convex quadratic problems. Due to the nature of the model in Zhang (2007), further literature review regarding branch-and-bound solution approach is provided as the background of the tailored nonlinear branch and bound implementation and the corresponding methods adapted to the model are illustrated as well in each section.

5.2 Background of Branch and Bound algorithm for MINLP

The Branch-and-Bound dates back to Land and Doig (1960). The first reference to nonlinear Branch-and-Bound can be found in Dakin (1965). Please refer to Borchers (2001) for major issues during the implementation of nonlinear Branch-and-Bound algorithms. The following subchapters will be organized in the way that differentiates the major issues in the implementation of nonlinear Branch-and-Bound.

5.2.1 Inner Nonlinear Programming Problems

The solving process for inner nonlinear programming problems can differentiate dramatically from problem to problem. Theoretically, it depends on the nature of the nonlinear problems --- the convexity of the problems. The nonlinear problem can be expressed as follows:

$$\begin{aligned} &\textbf{Minimize } f(x) \\ &\textit{subject to}: h(x) = 0; \\ &\qquad\qquad g(x) \le 0; \\ &\qquad\qquad x \in R^n \end{aligned}$$

If the both objective function f(x) and constraints g(x) are convex and h(x) is linear, then the nonlinear sub-problem is convex and relatively easy to solve. Both the original problem and the approximated problems are convex and especially, the latter one is a convex quadratic problem, therefore, we can comparatively easier get the optimal solution of the inner problem using both commercial and open source large scale nonlinear/quadratic packages. Currently, there are lots of algorithms available for convex nonlinear problem, especially for quadratic convex problem, including Generalized Reduced Gradient (GRG) (Gupta et al. 1985), Sequential Quadratic Programming (SQP), Interior Point Method, Penalty Function Methods (Fiacco and Mccormick 1968), and Active Set method specifically for quadratic programming. For small size problems, a modified simplex method combined with KKT conditions could be used to find the global optimum for convex quadratic problems as described in Winston (2004).

5.2.1.1 Generalized Reduced Gradient (GRG) and CONOPT

Gupta et al. (1985) pointed out the code based on Generalized Reduced Gradient algorithm has demonstrated significant superiority over the code based on other algorithms. It was originally developed by Abadie and Carpentier (1969) and Wolfe (1967). One of the major nonlinear solvers adopted in this research, GAMS-CONOPT, was developed based on GRG algorithm.

CONOPT is a solver for large-scale nonlinear optimization (NLP) developed and maintained by ARKI Consulting & Development A/S in Bagsvaerd, Denmark. It has been under continuous development for over 25 years. Based on the old proven GRG method, CONOPT has been a feasible path solver with many extensions. It has been designed to be efficient and reliable for a broad class of models. The original GRG method helps achieve reliability and speed for models with a large degree of nonlinearity, i.e. difficult models, and CONOPT is often preferable for very nonlinear models and for models where feasibility is difficult to achieve. Also, CONOPT has been designed for large and sparse models. Models with over 10,000 constraints are routinely being solved. Specialized models with up to 1 million constraints have also been solved with CONOPT. (www.conopt.com)

CONOPT guarantees global optimum for convex quadratic programming problems, therefore we can expect that CONOPT returns global optimum for our triangularly approximated inner problem.

CONOPT is recommended to be used along with a modeling system, such as GAMS, AMPL, LINDO Systems, TOMLAB optimization, and in this research it is used with GAMS as GAMS-CONOPT nonlinear solver.

5.2.1.2 Interior Point Method for Nonlinear Programming

Growing interest in efficient optimization methods has led to the development of interior-point or barrier methods for large-scale nonlinear programming. In particular, these methods provide an attractive alternative to active set strategies in handling problems with large numbers of inequality constraints (Wachter and Biegler, 2006). Another major nonlinear solver for inner quadratic problem is IPOPT, an open source interior point open source solver for large scale problems. The code has been written by Carl Laird and Andreas Wächter. IPOPT is designed to find the local optimum of nonlinear problems which can have both convex and non-convex objective functions and constraints as long as they are twice continuously differentiable. For convex quadratic problem, IPOPT also guarantee global optimum, which we are going to compare with other quadratic solvers such as CONOPT and CPLEX in next chapter.

The IPOPT distribution can be used to generate a library that can be linked to one's own C++, C, or FORTRAN code, as well as a solver executable for the AMPL modeling environment. Recently, it has also been successfully compiled as a nonlinear solver option under GAMS system through another open source implementation package GAMSlink (www.coin-or.org).

5.2.1.3 Iterative Algorithm for Inner Nonlinear Problem

The inner nonlinear problem we get after triangular approximations still remain non-quadratic format since the existence of variable y(k)s in the denominator. In order to simplify the inner problem into a pure quadratic problem, a simple iterative algorithm is implemented as follows:

New supplementary variables y1(k) are introduced.

Step 1: Initialization y1(k) := σ_k; # σ_k is the average demand of product k

Step 2: Solve the quadratic problem

If $\sum_1^k y(k) - \sum_1^k y1(k) < \varepsilon$; then stop; # ε is the stopping criterion

Otherwise, y1(k) := y(k); and repeat step 2.

Note: ε is set 0.01 for our testing experiments in next chapter.

5.2.2 Node Selecting Strategy in Nonlinear B&B

The choice of next sub-problem to be solved could have a significant impact upon the overall performance of the nonlinear Branch-and-Bound algorithm. In mixed integer programming, a variety of strategies are employed to select the next sub-problem to solve (Borchers, 2001).

Borchers (2001) also pointed out that one popular node selecting heuristic used in MILP known as "best bound rule" has also been widely used in nonlinear B&B. For "best bound rule", the sub-problem with the biggest upper bound is selected for maximization problems. This strategy has the advantage that the total amount of computation is minimized in a sense that once an integer solution is obtained, it will be good enough lower bound that will eliminate many nodes from consideration. However, it will consume more memory than other strategies. Due to the availability of large computer memory nowadays, the size of memory is no longer an issue; therefore, this strategy is employed in our implementation.

Another well known strategy is to branch from the newest node (Gupta and Ravindran, 1985). In this strategy, whenever a branching is carried out the nodes corresponding to the new problems are given preference over the rest of the unfathomed nodes. In another word, the node that is newest in the list of unfathomed nodes is selected for branching. This strategy is also known as the depth first approach, which has the advantage of saving storage space and relatively easy to implement. However, for larger scale problems, it usually takes much longer time for this strategy.

There are also other estimates or heuristics for node selection which not only considers the value of the objective function but also take into account the quality of the continuous optimal solutions, for example, the number of integer variables which are already integers in a node. Please refer to Gupta and Ravindran (1985) for detail.

5.2.3 Branching Strategy in Nonlinear B&B

There maybe a choice of several fractional variables to be branched once the node is selected. There are the following strategies as reported in Gupta and Ravindran (1985):

Branch the important integer variables first in a given model since it is possible to get some information of the important variables. This can be done by assorting the variables in a descending order according to their importance and branch those with lower index first. In the model, according to the special structure of the model, u(ijl)s are considered important binary variables over v(ij)s and w(j)s, and we found that v(ij)s and w(j)s can be relaxed as continuous variables given lower bounds of "0" and upper bounds "1", this is because of the fact that once u(ijl)s are determined, v(ij)s and w(j)s are easy to be fixed. Based on the facts above, we summarize the specific variable branching rules we tried for this model as follows:

- Select the ones furthest from being integers among u(ijl)s
- We do not intend to differentiate the branching priorities among u(ijl)s for different segments. This is because that only one segment for each component from each supplier will be allowed to select.

The choice that selecting those variables furthest from being integers is aimed at getting the largest degradation of the objective when branching is carried out so that more nodes can be fathomed at an early stage. The strategies mentioned above makes it unique and special in the implementation of the nonlinear B&B to the MINLP model.

Achterberg et al. (2005) provided a comprehensive review of a variety of state-of-the-art branching rules including strong branching, pseudo-cost branching and the hybrid of these two strategies. Based on these two strategies, they proposed a new generation of branching rule called reliability branching which has demonstrated superiority over other branching rules.

5.2.4 Obtaining Upper Bound for sub-problems

The branch and bound performance can also be improved by computing the upper bound of a sub-problem without actually solving the sub-problem. It is possible to get the upper bound of the optimal objective function value of the sub-problem from an optimal dual solution of a sub-problem's parent problem (Borchers, 1997). It is important to find a lower bound on MINLP objective as quickly as possible, which will eliminate some nodes from the early stage of branch and bound. The goal is to find an initial integer feasible solution. Gupta and Ravindran (1985) also proposed two heuristics to obtain the initial integer solutions.

Leyffer (2001) proposed a solution approach for MINLP by integrating SQP and Branch-and-Bound, in which SQP serves as the nonlinear solver. This algorithm does not require the NLP problem at each node to be solved to optimality. Instead, branching is implemented after each iteration of the NLP solving. Subsequently, the nonlinear problems are solved during the tree search process. The basic idea underlying the new approach is to branch early—possibly after a single QP iteration of the SQP solver.

5.3 Branch and Cut

In non-linear branch-and-cut approach, constraints called cutting planes are added into the nonlinear programming subproblems. These constrained are selected in a way that they reduce the size of feasible region of nonlinear programming subproblems without eliminating feasible solutions from consideration. By doing this, the possibility that the subproblem can be fathomed by bound is increased, moreover, the use of cutting planes makes it more likely that an integer solution will be obtained earlier in the branch and bound process. Several of cuts generating methods have been reported in Bienstock (1996), including mixed integer rounding cuts, knapsack cuts, intersection cuts.

5.4 Proposed Nonlinear B&B Algorithm

Based on the general framework of B&B in Wang and Sarker (2006), the proposed Branch-and-Bound algorithm specifically to the MINLP model in Zhang (2007) is illustrated as follows:

Step1: Solve the relaxed version of the problem (NLP), record its objective function value TC and set it the upper bound of the optimal solution of MINLP, Z_U=TC. An iterative algorithm is developed to transfer the nonlinear problem into a pure quadratic problem provided that the triangular approximation replaced the integration expression in the objective function. Set the upper bound and lower bound of v(ij) and w(j) to 1 and 0 respectively, relax them as binary variables.

Step2: If integer (binary) solutions are obtained, stop, otherwise set the lower bound of the problem $Z_L = -\infty$.

Step3: If there is any fractional value for U_{ijl}, choose the ones furthest from being integer variables and branch it, and get two sub-problems by adding $U_{ijl} = 1, U_{ijl} = 0$ one at a time. Solve the two sub-problems. Fathom the infeasible problem right away and keep the feasible solutions to form the nodes. If there is any integer (binary) solutions, fathom the corresponding node and update Z_L by setting Z_L equal to the new integer objective function value.

Step 4: Fathom the nodes by lower bound Z_L , if no nodes are available to fathom, and then stop, and the node with the objective function value Z_L is the optimal integer (binary) solution.

Step 5: Go to the node with better objective function value and go to **Step 3**.

5.5 Pseudo-Code

The branch-and-bound algorithm for MINLP problems with binary variables can be (Kalvelagen, GAMS):

```
{Initialization}
LB: =$-\infty$; UB: = $+\infty$; j=0;
Store root node (j=0) in waiting node list
while (waiting node list is not empty) do
  {Node selection}
 Choose the new generated nodes with the best objective function value from the waiting node list and remove it from the waiting node list
  Solve sub-problem
  if infeasible then
  Node is fathomed
  else if optimal then
    if integer solution then
      if obj > LB then
        {Better integer solution found}
        LB:= obj
        Remove nodes j from list with $UB_j < LB_j$
      end if
    else
      {Variable selection}
```

```
            Find variable $U_{ijl}$ furthest from being integer (binary) variables
            Create node $j_{new}$ with bound $U_{ijl} = 1$
            $UB_{j_{new}} := obj$
            Store node $j_{new}$ in waiting node list
            Create node $j_{new} + 1$ with bound $U_{ijl} = 0$
            $UB_{j_{new}+1} := obj$
            Store node $j_{new}$ in waiting node list
        end if
    else
        Stop: problem in solving subproblem
    end if
    $UB = max_j UBj$
end while
```

6. COMPUTATIONAL EXPERIMENTS OF NONLINEAR B&B ALGORITHM

In this chapter, the proposed nonlinear Branch-and-Bound algorithm is tested on the experimental tests of larger scale problems and the results of nonlinear B&B are also compared with the solutions obtained from open source based MINLP package BONMIN approaches. The focus is on nonlinear B&B, which is the specific controllable algorithm which is developed for Zhang (2007) and it is respected to provide with the optimal solutions. The comparison with BONMIN based approach will deliver some insights about the proposed nonlinear B&B and validate the results as well.

The nonlinear branch and bound algorithm has been implemented in GAMS, both GAMS-CONOPT and open-source IPOPT package are invoked to solve the inner convex quadratic problem in each node as illustrated in the pseudo code of chapter 5.

6.1 Validation of Nonlinear B&B algorithm

The same problem specification as Zhang (2007) is used to compare the proposed B&B algorithm with BONMIN based approach. The following table 8 shows the solution statistics:

Branch and Bound	Q=2000		
Criteria	BB-NLP(CONOPT)	BONMIN-CRTO	Difference
y_1	217.25	218.145	0.41%
y_2	178.24	178.163	0.04%
y_3	200.91	200.834	0.04%
y_4	191.28	191.438	0.08%

y_5	213.96	213.661	0.86%
$x_{1,4,2}$	2048.64	2049.38	0.04%
$x_{2,5,2}$	1773.00	1773.06	0.00%
$x_{3,1,2}$	2188.23	2191.01	0.13%
$x_{4,2,2}$	2444.25	2445.08	0.03%
$x_{5,3,2}$	2178.24	2178.16	0.00%
Objective Function Value	62445.12	62445.7	0.00%
Number of Nodes Visited	11	15	
Time Elapsed	2.722	0.37	
Number of Iterations	N/A	285	

Table 8: Validation of nonlinear B&B algorithm

BB-NLP (CONOPT) represents the approach of branch and bound algorithm with CONOPT as inner solver. BONMIN-CRTO represents the solution approach mentioned in chapter 4, which uses BONMIN branch and bound algorithm. The solutions of these two approaches indicate that the proposed tailored nonlinear branch and bound algorithm delivers fairly good solutions with little deviations from the BONMIN approach, this observation also validates the BB-CONOPT approach. It can also be noticed that in the new approach, less nodes are visited than BONMIN, which should be attributed to the specific branching strategy that is adopted.

6.2 Comparison of Different Nonlinear Solvers to Inner Quadratic Problem

As is mentioned in Chapter 5 about the motivation of developing nonlinear branch and bound algorithm for this kind of problems, it is important to make sure to get the global (or close to global) optimal solutions for the inner convex quadratic problems. Based on this consideration, different solvers available in GASM 22.5 student version are tried for the model in Zhang (2007). This is due to the fact that the size of the problem does not exceed the limit which is set for student version of GAMS and many nonlinear and quadratic solvers are available in the newest version of GAMS 22.5. The quadratic solvers include: BARON, COINIPOPT, CONOPT, CPLEX, MINOS, SNOPT, KNITRO, LGO, LINDOGLOBAL, MOSEK, MSNLP, OQNLP, PATHNLP, XPRESS. The following statistics in Table 9 shows clearly that the B&B algorithms with QCP solvers deliver exactly the same solution with different solving time. The results give the insight that both CONOPT and COINIPOPT will return the global optimal solutions for convex quadratic problems as it is illustrated in the solver's manual, and both of them will be used as the QP solver later for larger scale problems. A sample of output file for nonlinear B&B algorithm can be referred to APPENDIX E.

Q=2000, Nonlinear BB with different QCP solvers						
Criteria	CONOPT	COINIPOPT	MINOS	CPLEX	XPRESS	Difference
y_1	217.25	217.25	217.25	217.25	217.25	0%
y_2	178.24	178.24	178.24	178.24	178.24	0%
y_3	200.91	200.91	200.91	200.91	200.91	0%
y_4	191.28	191.28	191.28	191.28	191.28	0%
y_5	213.96	213.96	213.96	213.96	213.96	0%
$x_{1,4,2}$	2048.64	2048.64	2048.64	2048.64	2048.64	0%
$x_{2,5,2}$	1773.00	1773.00	1773.00	1773.00	1773.00	0%

$x_{3,1,2}$	2188.23	2188.23	2188.23	2188.23	2188.23	0%
$x_{4,2,2}$	2444.25	2444.25	2444.25	2444.25	2444.25	0%
$x_{5,3,2}$	2178.24	2178.24	2178.24	2178.24	2178.24	0%
O.B. Value	62445.12	62445.12	62445.12	62445.12	62445.12	0%

Table 9: Comparison of different nonlinear solvers to inner quadratic problem

6.3 Experiments on Larger Scale Problems

This research is aimed at providing good enough solution strategies to real world industrial application as initiated in Kim et al. (2002) and Zhang (2007). As the growing competition in the world wide PC industry nowadays, PC manufacturers have larger number of suppliers, final PC products, assembly components. For this consideration, reasonable size of larger scale problems are considered based on the cases from computer industries.

6.3.1 The Methodology of Generating Large Scale Problems

Based on the parameter specifications of the model in Zhang (2007), which can be accessed in APPENDIX C, ten more problems are randomly generated and tested under both nonlinear B&B approach in GAMS and BONMIN in AMPL. The parameters are randomly generated in a way that ranges from the corresponding parameters in the example of Zhang (2007) and are uniformly distributed with a fixed seed which makes it possible to obtain the exactly same problems later for the purpose of repetition of the testing substances.

Since the problem is originated from a real world application of a computer industry in Korean, the dimension of the larger scale problems being generated has also been tailored along with the consideration of current prevailing computer industry configurations. Dell^{TM} (www.dell.com) is taken as the example. It is indicated from the website that the procurement component in Dell^{TM} manages nearly 1.8 million purchase order lines per year from more than 5,000 suppliers worldwide. However, it must be admitted that in the real world supply chain application, there are so many other factors that are considered as well during supply chain and purchasing decision making process. It is neither necessary nor reasonable to try the problems with dimension as big as Dell has. Therefore, the models in Kim et al. (2002), Zhang and Ma (2007) and Zhang (2007) are just applicable to the decision making for a local decision making process.

The testing substances regarding different dimensions such as the number of suppliers, raw materials (components), final products and discount segments are listed below in table 10.

Suppliers	Products	Components	Segments
6	6	6	2
6	6	6	3
11	5	5	2
10	10	10	2
15	10	10	2
15	10	10	3
20	15	10	2
15	15	15	2
15	15	15	3
15	15	20	2

Table 10: Testing Substances

Randomly problems in AMPL format are generated. Since there is just minor change with respect to the input data format between AMPL and GAMS, only AMPL format is generated through C code. During the problem generating process, the followings factors are also considered:

- The value of Q (Manufacturing Capacity) is carefully selected to ensure the feasibility of different dimensions of problems. Q is estimated by using the corresponding information regarding y_k (the amount of product k to produce) and t_k (manufacturer's production consumption for unit of product k).
- The different values of available segments which should be set up by suppliers are carefully chosen to ensure the computing complexity of larger scale problems since the setting of segments has certain impact over the duality gap between the relaxed solutions and the final integer solutions.

6.3.2 Discussion of Results to Larger Scale Problem

The main contribution of this research is that the original MINLP problem with integration item in objective function and the nonlinear part of square roots in denominator is successfully transferred into a MIQCP (Mixed Integer Quadratic Constrained Problem). As the B&B approach has been validated in the previous part of 6.1, the solutions of the MIQCP can be trusted.

In this section, larger scale problems are tested to illustrate the robustness of nonlinear B&B algorithm. Both IPOPT and CONOPT are employed as the inner quadratic solvers to ensure the optimum and here only covering range triangular approximation is provided to simplify the process. It must be pointed out here that the inner problem in BONMIN-B&B and GAMS-SBB in each node is not a quadratic problem but indeed a convex nonlinear problem. The results from 10 randomly generated problems in nonlinear B&B,

GAMS-SBB and AMPL-BONMIN are reported. In the following tables, NBB represents the branch and bound algorithm coded in GAMS environment, SBB represents GAMS-SBB MINLP solver, while BBB represents open source MINLP branch and bound algorithm BONMIN and BOA means Outer Approximations algorithm option that is selected in BONMIN to solve the MINLP. N-S-GAP shows the gap between NBB and SBB while N-B-GAP represents the gap between NBB and BBB. The number of nodes visited gives the number of nonlinear problems that have been solved by nonlinear solvers.

Note: In order to make the testing substances easy to recognize, a notation system is introduced to standardize different substances. It can be generally expressed as "s(n)-p(m)-c(q)-seg(d)", among which letters "n", "m", "q", "d" represents the number of suppliers, final products, components and available segments respectively. For example, s6-p8-c10-seg2 represents the substance with 6 suppliers, 8 final products, 10 components and 2 available segments in total.

	NBB	SBB	BBB	N-S-GAP (%)	N-B-GAP (%)
O.B. value	248393.3	248397	248395	0.00	0.00
x(1,2,1)	777.49	783.17	783.177	0.73	0.73
x(1,4,2)	1600	1600	1599.96	0.00	0.00
x(2,2,2)	1648.81	1656.62	1656.59	0.47	0.47
x(2,5,2)	1500.01	1500.01	1500.01	0.00	0.00
x(3,3,2)	2712.11	2718.5	2718.46	0.24	0.23
x(4,1,2)	2577.36	2583.73	2583.69	0.25	0.24
x(5,5,2)	2439.41	2444.33	2444.29	0.20	0.20
x(6,1,2)	2408.81	2414.36	2414.32	0.23	0.23
y(1)	227.29	227.94	227.945	0.29	0.29
y(2)	164.45	164.62	164.617	0.10	0.10
y(3)	222.12	222.39	222.389	0.12	0.12
y(4)	209.41	210.18	210.178	0.37	0.37
y(5)	195.97	196.76	196.759	0.40	0.40

y(6)	178.44	178.58	178.581	0.08	0.08
nodes visited	95	213	56	N/A	N/A
CPU time(s)	15.049	6.094	2.73	N/A	N/A

Table 11: Statistics for s6-p6-c6-seg2

From table 11, it is easy to see the substantial improvement from SBB to NBB with respect to the number of nodes visited and the gaps are very small regarding both objective function values and the variable solutions. As can be observed here that BBB delivers the best performance, and BBB is taken as the reference to ensure the optimality of the proposed branch and bound algorithm. However, it is only fair to compare NBB with SBB since they are based on the same platform --- GAMS.

	NBB	SBB	BBB	N-S-GAP (%)	N-B-GAP (%)
O.B. value	271620.82	271620.94	271619	4E-05	7E-04
x(1,2,3)	2500.01	2500.01	2500.01	0E+00	0E+00
x(2,2,1)	459.99	459.99	459.952	0E+00	8E-03
x(2,5,3)	2724.99	2724.99	2724.96	0E+00	1E-03
x(3,3,3)	2740.12	2740.13	2740.06	4E-04	2E-03
x(4,1,3)	2599.99	2599.99	2599.93	0E+00	2E-03
x(4,3,1)	6.59	6.77	6.761	3E+00	3E+00
x(5,5,3)	2500.01	2500.01	2500.01	0E+00	0E+00
x(6,1,3)	2500.01	2500.01	2500.01	0E+00	0E+00
y(1)	229.45	229.4	229.392	2E-02	3E-02
y(2)	165.09	165.11	165.106	1E-02	1E-02
y(3)	225.19	225.35	225.353	7E-02	7E-02
y(4)	215.41	215.45	215.448	2E-02	2E-02
y(5)	197.27	197.22	197.215	3E-02	3E-02
y(6)	179.37	179.36	179.364	6E-03	3E-03
nodes visited	129	277	86	N/A	N/A
CPU time(s)	96.887	9.838	6.17	N/A	N/A

Table 12: Statistics for s6-p6-c6-seg3

Table 12 and 13 demonstrates the results of another two examples, compared with BONMIN-B&B algorithm, the solution obtained from CONOPT as the inner quadratic solver is validated again, and the huge improvement can be observed regarding the number of nodes visited between the tailored B&B algorithm coded in GAMS and GAMS-SBB provided that same solver CONOPT is employed to solve the inner nonlinear problems.

	NBB	SBB	BBB	N-S-GAP (%)	N-B-GAP(%)
O.B. value	386590	386608.4	386608	0.00	0.00
x(1,6,2)	2000	2000	2000	0.00	0.00
x(1,9,2)	2150.16	2168.5	2168.5	0.85	0.85
x(2,2,2)	2000	2000	2000	0.00	0.00
x(2,3,2)	2000	2000	2000	0.00	0.00
x(3,8,2)	4222.22	4240.99	4240.99	0.44	0.44
x(4,5,2)	3606.19	3621.75	3621.75	0.43	0.43
x(5,1,2)	2374.32	2394.4	2394.4	0.84	0.84
x(5,4,2)	2000	2000	2000	0.00	0.00
x(6,2,1)	523.47	535.09	535.09	2.17	2.17
x(6,3,2)	2200	2200	2200	0.00	0.00
x(7,1,2)	2000	2000	2000	0.00	0.00
x(7,5,2)	2029.72	2047.7	2047.7	0.88	0.88
x(8,9,2)	3777.34	3794.26	3794.26	0.45	0.45
x(9,1,2)	2706.83	2718.93	2718.93	0.45	0.45
x(10,6,2)	3600	3600	3600	0.00	0.00
x(10,9,1)	27.29	42.35	42.35	35.56	35.56
y(1)	215.96	216.56	216.563	0.28	0.28
y(2)	175.03	175.85	175.849	0.47	0.47
y(3)	203.38	204.64	204.636	0.62	0.61
y(4)	206.26	207.08	207.076	0.40	0.39
y(5)	206.33	206.87	206.872	0.26	0.26
y(6)	215.57	216.76	216.763	0.55	0.55
y(7)	246.16	247.24	247.238	0.44	0.44
y(8)	146.48	146.63	146.635	0.10	0.11
y(9)	159.43	160.24	160.24	0.51	0.51
y(10)	175.81	176.86	176.865	0.59	0.60
nodes visited	291	138	901	N/A	N/A
CPU time(s)	189.818	280.535	13.82	N/A	N/A

Table 13: Statistics for s10-p10-c10-seg2

For the rest of the testing substances, only the objective functions and the number of nodes visited of the three solution approaches will be compared, and they are illustrated in Table 14. Please refer to Appendix F for the detailed solutions for CONOPT solvers, the results of all solvers can be found in the attached CD. For BONMIN B&B approach, B&B is first tried, if it failed then OA (Outer Approximation) algorithm is tried again, however if OA failed to converge within 2 days, then it is stopped manually.

	Objective Function Value			Nodes Visited		
	NBB	SBB	BBB	NBB	SBB	BBB
s11-p5-c5-seg2	65613.4	65613.95	65778.1	43	72	27
s15-p10-c10-seg2	55693	55745.51	55744.9	555	406	374
s15-p10-c10-seg3	74492.33	74533.81	74533.3	12919	33110	15594
s20-p15-c10-seg2	60712.74	60767.56	60766.9	71	114	64
s15-p15-c15-seg2	519132	519134	519116 (BOA)	1073*	4193(4098)	N/A
s15-p15-c15-seg3	565459.8	565462	565447(BOA)	779*	20692(15594)	N/A
s15-p15-c20-seg2	453753.3	453777	453753	1981*	7003	7780

	CPU time (s)	
	NBB	SBB
s11-p5-c5-seg2	5.6	8.617
s15-p10-c10-seg2	486.62	196.8
s15-p10-c10-seg3	17131	31535.99
s20-p15-c10-seg2	28.12	314.02
s15-p15-c15-seg2	815.69	7254.339
s15-p15-c15-seg3	1054.26	38227.13
s15-p15-c20-seg2	2973.75	31423.65

Table 14: Simplified statistics for the rest of the examples

It must be pointed out that in Table 14, the value marked with star "*" means that BONMIN-B&B fails to solve the problem which is solved by BONMIN-OA successfully. In order to simplify the branch and bound process, the binary variables for higher discount segments are fixed based on the corresponding solutions we get from BONMIN-OA.

In the case of s15-p15-c15-seg2 and s15-p15-c15-seg3, under the same node selecting rule which is best bound selection, it can be noticed that the specific variable branching strategy for this model has demonstrated much more superior performance regarding the number of nodes visited than SBB. Very close objective function values are obtained as well and less CPU time is used for NBB than SBB when the problem becomes larger. The slight difference between the objective values could be attributed into the iterative algorithm technique which is introduced for the inner problem.

As the motivation of collaborating with open-source solvers, COIN-OR nonlinear solver IPOPT is installed in GAMS by compiling a project package GAMSlink in COIN-OR. Detailed procedures and instructions in terms of GAMSlink installation can be referred to https://projects.coin-or.org/GAMSlinks according to various operating platforms. The following table 15 illustrates the comparison between IPOPT and CONOPT as the inner solvers to quadratic problems. The integer tolerance of 1e-4 is enforced when IPOPT is used as the solver since it is an external solver installed under GAMS while CONOPT comes with a commercial GAMS nonlinear solver.

	GAMS-IPOPT			GAMS-CONOPT		
Testing Instances	O.B. Value	Nodes visited	CPU time	O.B. Value	Nodes visited	CPU time
s5-p5-c5-seg2	62445.12	11	6.858	62445.12	11	1.546
s6-p6-c6-seg2	248393.32	133	11040.745	248393.3	95	15.049
s6-p6-c6-seg3	271620.82	127	95.586	271620.82	129	96.887
s10-p10-c10-seg2	386589.98	299	469.669	386590	291	189.818

s11-p5-c5-seg2	65613.4	29	32.783	65613.4	43	5.6
s15-p10-c10-seg2	55693.48	721	12803.811	55693	555	486.62
s15-p10-c10-seg3	74492.33	3017	89669.691	74492.33	12919	17131
s20-p15-c10-seg2	60712.74	119	383.164	60712.74	71	28.12
s15-p15-c15-seg2	Fail to solve	n/a	n/a	519132	1073*	815.69
s15-p15-c15-seg3	n/a	> 65000	n/a	565459.8	779*	1054.26
s15-p15-c20-seg2	453770.19	8578*	44951.617	453753.3	1981*	2973.75

Table 15: Comparison of IPOPT and CONOPT as inner solver

The result in Table 15 shows that IPOPT and CONOPT deliver exactly the same objective function values while CONOPT generally visited less number of nodes and CPU time before reaching the optimal solutions, which should be attributed to the better integration of CONOPT with GAMS. However, IPOPT as the open source nonlinear solver has more flexibility in other platforms, such as AMPL or C/C++ code; it has greater potential to be explored in future research.

It has to be admitted that if the dimensions of the problem are increased further, the computing time increases exponentially for most of the cases although some randomly generated problems have smaller duality gap itself which will make it easier to solve.

6.4 Discussion of Complexity of Nonlinear Branch-and-Bound

To analyze the complexity of B&B algorithm, N is denoted as the number of binary variables in the MINLP problem, and the number of nodes can be expressed as a function of N. It is shown in a very similar problem defined in Wang (2006) that the complexity of the B&B algorithm is $(N^{N+1}-1)/(N-1) \approx O(N^N)$. And the worst case of the B&B algorithm would be the exponential form. When N becomes bigger and computing time will soon be intractable. For the model in Zhang (2007), the number of different combinations of the binary solutions could be as many as 2^N, it could be extremely hard to solve when N is very big. The actual computing time depends on the duality gap of the problem which is determined by the specifications of different problems. In other word, different B&B strategies can decrease the computational time to some extend; however, the running time of the worst case could be intractable.

6.5 Idea of Heuristics

6.5.1 Heuristic One

A small heuristics is proposed here as part of the effort to deal with the situation where large amount of binary variables appears, especially for u_{ijl}. The general idea can be categorized into two phases. In phase one, the only remaining binary variables will be w_j. Based on the results from phase one, certain amount of binary variables of U_{ijl} will be fixed accordingly. Then the algorithm goes to phase two, where the regular nonlinear B&B algorithm applies as discussed previously. The heuristic algorithm is demonstrated as following:

PHASE I

Step 1: Eliminate the set of all segments, binary variables u_{ijl}, v_{ij}, and the corresponding constraints involving u_{ijl} and v_{ij}, replace the variables x_{ijl}, du_{ijl} and dl_{ijl} by x_{ij}, du_{ij} and dl_{ij} respectively. Use the last segment of parameters of du_{ijl} for the new parameters du_{ij} and use the parameters of first the segment of dl_{ijl} for dl_{ij}. Eliminate parameters c_{ijl} and replace them with c_{ij} with the values of the highest segment of c_{ijl}. (By following the procedures in step 1, the model become a MINLP with the only binary variables w_j, which means much less number of binary variables)

Step 2: Solve the MINLP from step 1 with the Nonlinear B&B algorithm, and record the solution. Select the solutions x_{ij} with the values greater than the last segment limit in the original problem, and set the corresponding u_{ijl} in the original problem to "1".

Phase II

Solve the original problem using the tailored Nonlinear B&B algorithm with some of the u_{ijl} fixed as indicated in the Step 2 of Phase I and record the solution. The solution can be considered as a heuristic solution of the original problem. In addition, this heuristic solution could act as the best found solution (lower bound).

6.5.2 Heuristic Two

This heuristic idea focuses on providing an initial integer solution quickly enough as the lower bound before the exact nonlinear B&B search actually starts. The heuristic shares the same first step as the Step 1 in Heuristic 1. After getting the objective value of the problem in Step 1, it is recorded as OBJ1. The variables, whose values are not within the limits of the last segments, are selected, then based on the actually values of these variables, they are assigned to the corresponding segments they belong to and set the associated binary variables u_{ijl} equal to "1". The integer solution objective value can be obtained by using the following formula:

Best_found = OBJ1 - $\sum x_{ij} * \left(c_{ijl} - c_{ij}\right)$ for all x_{ij} that are less than the lower limit of the highest segment in the original problem.

Then this Best_found value can be one integer solution which can serve as the initial lower bound of the exact nonlinear B&B algorithm.

For very large scale problems, although Branch-and-Bound algorithm can provide exact solutions, the time it takes for optimum is highly intractable; the solution from Phase I can be simply used to construct a heuristic solution by assigning the values of x_{ij} to its corresponding segments based on the limits of different segments. Under this scenario, the definitions of the segments have the final say regarding the quality of the solutions.

	Heuristic 1		NBB	
	o. b. value	nodes visited	o. b. value	nodes visited
s5-p5-c5-seg2	62445.12	11	62445.12	11
s6-p6-c6-seg2	248393.32	41	248393.3	95
s6-p6-c6-seg3	270988.82	46	271620.82	129
s10-p10-c10-seg2	386589.98	96	386590	291
s15-p15-c15-seg2	519132.04	660	519132	1073

Table 16: Comparison between NBB and Heuristic 1

Table 16 shows the comparison of the selected examples between Heuristic 1 and nonlinear branch-and-bound algorithm, in both of which CONOPT is used as the inner solver. It can be easily seen that except for the case s6-p6-c6-seg3, the proposed Heuristic 1 delivers exact objective values with less nodes visited. However, it has to be admitted that Heuristic 1 can not guarantee the optimal solution and when the dimension of the problem becomes bigger. With limited number of higher segments fixed, it still takes longer time to solve. A more approximated solution can be obtained through Heuristic 2, in which the number of binary variables equals to the number of suppliers and it is straight forward for much larger scale problems.

7. CONCLUSIONS AND RECOMMENDATIONS

7.1 Conclusions

In this thesis, a series of new solution approaches to a class of supply chain problems which share the similarity of having continuously distributed uncertain demand are developed. This class of stochastic supply chain problems can be represented by three major decision problems introduced by Kim et al. (2002), Zhang and Ma (2007) and Zhang (2007) separately. The model in Zhang (2007) is the most difficult and representative one among them. The major difficulties dealing with this class of problems include two aspects:

- ❖ How to deal with the integration functions that appear in the formulation to get reliable solutions of the nonlinear problem
- ❖ How to handle the Branch-and-Bound procedure efficiently when different quantity discount schemes are introduced by various suppliers, and make the supplier selection decision.

These two questions are addressed and explored deeply by focusing on the problem in Zhang (2007). Nonlinear B&B algorithm is considered as the frame work of the solution approaches, therefore this research work can be eventually divided into two portions: the effort to solve each problem in each node and branch-and-bound strategy. Before addressing these efforts, an AMPL-external function based approach is developed which is aimed to provide a solution from BONMIN by introducing self-defined external functions to tackle the integration parts. However, the optimality of this approach can not be guaranteed and it can not handle larger scale problems although it should return optimal solutions since the problem is convex problem by the manual of BONMIN.

In the effort to deal with inner nonlinear problem, intensive literature researches are conducted in Newsboy or Newsvendor problems due to the similarities they share in the objective functions. Specifically, inspired by Areeratchakul and Abdel-Malek (2006) triangular approximations are employed to transfer the integration function into quadratic functions. This transformation makes it possible to solve larger scale problems. The triangular approximation approach is validated by testing all the example problems in both Kim's and Zhang's paper and the results indicate that this approach indeed delivers fairly good approximation for real world applications. The quadratic form of the objective function combined with linear constraints makes it possible to work on larger scale problems. Both open source MINLP BONMIN and commercial GAMS-SBB are used to solve the problems and the results are reported as well.

Further more, a tailored branch and bound algorithm is introduced and implemented in GAMS which is the core work of this research. An iterative algorithm is proposed to eliminate the variables in denominators about inventory and setup cost. By doing this, the problem becomes a pure quadratic problem in each node and both open-source nonlinear solver IPOPT and GAMS nonlinear solver CONOPT are employed to solve the inner problems. Based on the specific structure of this problem, a tailored variable branching rule is used when a non-integer solution is obtained after solving a node and has demonstrated significant improvements regarding the number of nodes visited compared with SBB.

7.2 Contributions

The main contributions of this research include the following aspects:

- By taking advantage of the similarity shared with newsboy model, the triangular approximation approaches are applied to the integration part in the objective functions instead of using Lagrange Multiplier based approaches. A quadratic function is obtained to replace the integral functions by decomposing it. Based on quadratic functions, future manipulations are possible to make along with modifications of the model by considering different other factors.
- Open source packages such as BONMIN and IPOPT e t. are compiled and used in this research as part of the solutions approaches, which could be intensively explored later.
- An iterative algorithm is proposed to transfer the inner nonlinear problem into a pure quadratic problem. This makes it possible for some future work aiming at providing research on the solution of the quadratic problems.
- Heuristics are developed for possible larger scale problems and the results of selected problems are also reported.
- A nonlinear branch and bound frame work tailored specifically for quantity discount segments has been successfully built and tested. For nonlinear B&B, both open-source solver COIN-IPOPT and commercial solver CONOPT in GAMS are used as the inner problem solver. Through comparisons between CONOPT and other quadratic solvers in GAMS, it can be seen that CONOPT delivers global optimum for the inner convex quadratic problem. Therefore the solution obtained from nonlinear B&B approach. At the same time, GAMS-SBB is also employed to solve the MINLP and the results are compared with proposed nonlinear B&B. Since there are a lot of considerations in the process of branch

and bound as discussed in Chapter 5, this frame work successfully build a platform for future research to this kind of problems with discount segments.

7.3 Recommendations to Future Research

As is discussed previously, this work can be continued in many possible ways. Specially, the following considerations could be considered as extensions:

- More accurate approximations of the integral part could be explored, please refer to Abdul-Malek and Areeratchakul (2007) for information.
- The specific algorithm for inner quadratic problem after approximation could be developed to ease the solving process.
- For inner nonlinear problem, the exact and more efficient algorithm would be expected and this should be jointly considered with the newest development of solutions approaches to Newsstand models.
- Open-source solver packages may be used in a more flexible way if good algorithms for inner problems are available.
- Possible improvements can be made to tighten the bound and relaxed solution in each node, such as Branch-and-Cut and other related techniques.

APPENDIX

APPENDIX A: Kim's case 1

Table 17: Product specification - input requirements g(i,k)

Product	Component(i)				
	Component1	Component 2	Component 3	Component4	component5
Product1	1	0	1	1	0
Product2	0	1	1	0	1

Table 18: Supply Costs (c(i,j) and q(i))

Supplier	Capacity (qi)	Component (i)				
		Component1	Component2	Component3	Component4	Component5
Supplier1	39.4	115	285	155		
Supplier2	34.6			135		
Supplier3	36.7			147		
Supplier4	41.8				171	181

Table 19: Production Information

Product	r(k)	t(k)	μ(k)	σ(k)	a(k)	b(k)
Product1	525	80	25.1	3.972	10	100
Product2	720	80	25.15	3.747	12	170

APPENDIX B: Kim's case 2

Table 20: Product-related Parameters

k	1	2	3	4	5
r(k)	150	200	220	230	250
t(k)	1	2	2	2	3
μ(k)	200	160	180	160	200
σ(k)	80	60	70	60	80
a(k)	100	90	50	90	150
b(k)	60	40	20	10	100

Table 21: Input Requirements g(i,k)

i/k	product1	product2	product3	product4	product5
component1	2	1	3	1	3
component2	1	3	2	1	2
component3	3	2	1	4	1
component4	2	1	2	3	4
component5	1	3	2	2	3

Table 22: Supply Costs c(i,j)

i/j	supplier1	supplier2	supplier3	supplier4	supplier5
component1	8	8	12	6	15
component2	10	15	8	10	5
component3	5	7	14	9	8
component4	9	5	10	13	8
component5	12	9	5	7	6

Table 23: Resource usage of supplier j per unit of component i produced n(i,j)

i/j	supplier1	supplier2	supplier3	supplier4	supplier5
component1	1.5	2	3	1	3
component2	2	1	1	3	1
component3	2	1.5	1	3	2.5
component4	1.5	3	2.5	2	3
component5	3	2	3	2	1.5

Table A-8 Capacity of supplier (q_j)

Table 24: Capacity of supplier q(j)

j	supplier1	supplier2	supplier3	supplier4	supplier5
q(j)	10000	7500	9000	6000	12500

APPENDIX C: Zhang's case

On top of Kim's case two, Zhang (2007) added

Table 25: Setup cost for order component i from supplier j (ksetup(i,j))

i/j	supplier1	supplier2	supplier3	supplier4	supplier5
component1	150	300	200	200	200
component2	200	100	100	300	100
component3	200	150	100	300	250
component4	150	300	250	200	300
component5	300	200	300	200	150

Table 26: Unit hold cost for component i from supplier j, (h(i,j))

i/j	supplier1	supplier2	supplier3	supplier4	supplier5
component1	1.5	1.5	1.5	1.5	1.5
component2	1.6	1.6	1.6	1.6	1.6
component3	1.6	1.6	1.6	1.6	1.6
component4	1.6	1.6	1.6	1.6	1.6
component5	1.6	1.6	1.6	1.6	1.6

Table 27: Unit prices of component i from supplier j on price segment 1 (c(ij1))

i/j	supplier1	supplier2	supplier3	supplier4	supplier5
component1	8	8	12	6	15
component2	10	17	8	10	5
component3	5	7	14	9	8
component4	9	5	10	13	8
component5	12	9	5	7	6

Table 28: Unit prices of component i from supplier j on price segment 2 (c(ij2))

i/j	supplier1	supplier2	supplier3	supplier4	supplier5
component1	6.5	6	10	5	11
component2	8	14	6	8	5
component3	4	6	11	7	6
component4	7	4	8	10	4.5
component5	10	8	4	6	5.5

Table 29: Segments specification

	Upper Limit	Lower Limit
segment1	1000	0
segment2	10000	1000.001

APPENDIX D: Results for Kim's case 2

Table 30: Results for Kim's case 2 using E-CRTO

Q	1500	1550	1600	1650	1700	1750	1800	1850	1900	1950	2000	2050	2100	2150	2200
y1	190.6	193.733	196.866	199.999	203.133	206.266	209.399	212.533	215.666	218.799	221.932	223.03	226.163	229.296	231.143
y2	132.191	136.606	141.021	145.436	149.851	154.266	158.681	163.096	167.511	171.926	176.342	183.131	187.546	191.961	193.786
y3	143.854	149.716	155.577	161.438	167.3	173.161	179.023	184.884	190.745	196.607	202.468	209.161	215.023	220.884	224.339
y4	150.249	154.664	159.079	163.494	167.909	172.324	176.74	181.155	185.57	189.985	194.4	197.694	202.109	206.524	209.32
y5	152.271	158.098	163.926	169.754	175.582	181.41	187.238	193.066	198.894	204.722	210.55	215.666	221.494	227.322	230.928

APPENDIX E: A sample of the nonlinear B&B output

```
COMPILATION TIME    =      0.004 SECONDS    3 Mb  LEX225-148 May 29, 2007

^LGAMS Rev 148  x86_64/Linux                                  12/14/07 11:59:47
Page 8

Final Model for Discount Model with case 2

E x e c u t i o n

----   494 PARAMETER bblog  logging information

          node      ub   solvestat  modelstat     obj      integer     best   waiting

node1     1.00     +INF      1.00       2.00   63263.77                        2.00

node2     2.00  63263.77      1.00       2.00   62190.92                        3.00

node3     3.00  63263.77      1.00       2.00   63256.06                        4.00
```

```
node4      6.00   63256.06     1.00     2.00   59573.80                         5.00

node5      7.00   63256.06     1.00     2.00   63059.41                         6.00

node6     10.00   63059.41     1.00     2.00   60391.22                         7.00

node7     11.00   63059.41     1.00     2.00   62963.53                         8.00

node8     14.00   62963.53     1.00     2.00   60263.90                         9.00

node9     15.00   62963.53     1.00     2.00   62733.07                        10.00

node10    18.00   62733.07     1.00     2.00   61672.81                        11.00

node11    19.00   62733.07     1.00     2.00   62445.12     1.00      1.00

----   499 PARAMETER bestu  record best solution

                  segm2

comp1.supp4       1.00

comp2.supp5       1.00

comp3.supp1       1.00

comp4.supp2       1.00

comp5.supp3       1.00

----   499 PARAMETER bestv  record best solution

            supp1     supp2     supp3     supp4     supp5

comp1                                      1.00

comp2                                                1.00

comp3        1.00

comp4                  1.00

comp5                            1.00

----   499 PARAMETER bestw

supp1 1.00,   supp2 1.00,   supp3 1.00,   supp4 1.00,   supp5 1.00

----   499 PARAMETER bestx

                 segm2

comp1.supp4    2048.64

comp2.supp5    1773.00

comp3.supp1    2188.23
```

```
comp4.supp2    2444.25

comp5.supp3    2178.24

----   499 PARAMETER besty

prod1 217.25,    prod2 178.24,    prod3 200.91,    prod4 191.28,    prod5 213.96

----   499 PARAMETER bestfound          =     62445.12  lowerbound in B&B tree

EXECUTION TIME      =       1.661 SECONDS      4 Mb  LEX225-148 May 29, 2007

USER: Guoqing Zhang                          G070507:1625AP-LNX

    University of Windsor, Industrial and Manufacturing SystemsDC6434

    License for teaching and research at degree granting institutions

**** FILE SUMMARY

^LGAMS Rev 148  x86_64/Linux                                          12/14/07 11:59:47
Page 9

Final Model for Discount Model with case 2

E x e c u t i o n

Input      /home/chen11q/GAMS/BB_debug.gms

Output    /home/chen11q/GAMS/BB_debug.lst
```

APPENDIX F: Detailed output of nonlinear B&B for testing substances

Please refer to the attached CD for the comprehensive information about nonlinear B&B, GAMS-SBB and AMPL-BONMIN. The following shows CONOPT as inner solver.

Table 31: A, B, C calculation of s10-p10-c10-seg2

ak	bk	rk	Mean	Std.Dev	Ak	Bk	Ck	delta(k)	y(l,k)
130	70	300	210	20	-3.90	1818.64	-150861.9	0.02	177.96
125	95	360	160	30	-3.02	1160.50	-57808.14	0.01	111.94
125	90	380	180	50	-1.86	876.06	-41035.24	0.01	99.90
105	30	220	200	20	-2.77	1255.53	-99146.63	0.02	167.96
115	105	340	190	30	-2.91	1281.99	-80542.34	0.01	141.94
80	40	420	180	50	-1.69	836.76	-31221.90	0.01	99.90
120	10	480	190	50	-1.90	1018.50	-45797.10	0.01	109.90
55	105	480	140	10	-9.99	3011.61	-161225.8	0.03	123.98
100	35	320	140	30	-2.37	855.24	-34008.30	0.01	91.94
75	55	380	160	30	-2.65	1048.97	-45245.09	0.01	111.94

Table 32: Detailed solution of s11-p5-c5-seg2

x(142)	2098.26	y1	220.4
x(252)	1817.82	y2	183.1
x(312)	2236.15	y3	206.58
x(462)	2502.58	y4	195.62
x(572)	2233.1	y5	219.66
CPU time (s)	5.602	nodes	43

Table 33: A, B, C calculation of s15-p10-c10-seg2

ak	bk	rk	Mean	Std.Dev	Ak	Bk	Ck	delta(k)	y(l,k)
100	60	150	200	80	-0.6	336.9	-23121	0.0039	71.8
90	40	200	160	60	-0.9	399.7	-17903	0.0052	63.9
50	20	220	180	70	-0.6	357.8	-11978	0.0045	67.9
90	10	230	160	60	-0.9	429.7	-17903	0.0052	63.9
150	100	250	200	80	-1.0	540.2	-35035	0.0039	71.8
120	50	265	210	65	-1.0	606.2	-36907	0.0048	105.9
80	75	250	150	75	-0.8	380.3	-12751	0.0042	29.9
100	55	180	170	70	-0.7	366.4	-19501	0.0045	57.9
95	70	200	190	65	-0.9	445.5	-24512	0.0048	85.9
60	65	280	175	60	-1.1	506.2	-17055	0.0052	78.9

Table 34: Detailed solution of s15-p10-c10-seg2

x(1,4,2)	1000	prod1	183.01
x(1,15,2)	2378.92	prod2	162.09
x(2,5,2)	2944.54	prod3	152.06
x(3,1,2)	3211.32	prod4	147.68
x(4,6,2)	1500	prod5	192.31
x(4,9,2)	2658.24	prod6	221.08
x(5,3,2)	1461.91	prod7	158.29
x(5,7,2)	1800	prod8	171.49
x(6,3,2)	1165.3	prod9	182.81
x(6,10,2)	1919.46	prod10	181.26
x(7,7,2)	1000	CPU time (s)	486.62
x(7,8,2)	2545.68	nodes	555
x(8,4,2)	3044.79		
x(9,10,2)	4161.08		
x(10,2,2)	1875.66		

x(10,6,2)	1000	

Table 35: Detailed solution of s15-p10-c10-seg3

x(1,4,3)	2536.57	y1	192.47
x(1,15,2)	1000	y2	168.87
x(2,1,3)	2745.84	y3	164.34
x(2,5,1)	328.9	y4	158.91
x(3,11,3)	3371.43	y5	200.96
x(4,2,3)	2345.88	y6	227.34
x(4,13,3)	2000	y7	164.7
x(5,3,3)	2202.13	y8	179.51
x(5,7,2)	1216.48	y9	187.51
x(6,3,3)	2239.09	y10	187.34
x(6,10,2)	1000	x(8,4,3)	3179.15
x(7,7,2)	1729.4	x(9,11,3)	4368.72
x(7,15,3)	2000	x(10,1,3)	3005.55
CPU time (s)	17131	nodes	12919

Table 36: Solution of s15-p15-c15-seg2

x(1,5,2)	4452.8	y1	161.51
x(2,4,2)	5032.23	y2	146.21
x(3,3,2)	4736.09	y3	172.45
x(4,2,2)	2366.67	y4	162.33
x(4,5,2)	2272.55	y5	155.79
x(5,1,2)	4776.15	y6	192.41
x(6,6,1)	571.44	y7	218.41
x(6,7,2)	3799.99	y8	207.23
x(7,6,2)	2000.01	y9	184.52
x(7,10,2)	2196.24	y10	163.51
x(8,1,2)	2000.01	y11	165.95
x(8,9,2)	3081.67	y12	166.44
x(9,6,2)	4524.85	y13	148.67
x(10,8,2)	4532.05	y14	161.24
x(11,15,2)	5224.63	y15	128.59
x(12,3,2)	2178.95	x(14,11,1)	445.69
x(12,14,2)	2994.75	x(14,12,2)	3120
x(13,3,1)	142.35	x(15,8,2)	2000.01
x(13,11,2)	4077.16	x(15,13,2)	2934.22

x(14,7,2)	2000.01	CPU time(s)	815.69

Table 37: A, B, C calculation of s15-p10-c10-seg2

ak	bk	rk	Mean	Std.Dev	Ak	Bk	Ck	delta(k)	y(l,k)
100	60	150	200	80	-0.60	337	-23121	0.004	71.8
90	40	200	160	60	-0.86	400	-17903	0.005	63.9
50	20	220	180	70	-0.65	358	-11978	0.004	67.9
90	10	230	160	60	-0.86	430	-17903	0.005	63.9
150	100	250	200	80	-0.98	540	-35035	0.004	71.8
120	50	265	210	65	-1.04	606	-36907	0.005	105.9
80	75	250	150	75	-0.84	380	-12751	0.004	29.9
100	55	180	170	70	-0.75	366	-19501	0.004	57.9
95	70	200	190	65	-0.88	446	-24512	0.005	85.9
60	65	280	175	60	-1.05	506	-17055	0.005	78.9
110	45	200	180	80	-0.69	382	-21661	0.004	51.8
85	50	150	175	60	-0.74	352	-19488	0.005	78.9
160	60	130	200	65	-0.84	451	-39724	0.005	95.9
90	65	220	150	70	-0.84	373	-14699	0.004	37.9
75	55	180	140	75	-0.65	281	-10754	0.004	19.9

Table 38: Solution of s20-p15-c10-seg2

y1	183.02	x(1,4,2)	2062.45	x(10,18,2)	4315.86
y2	158.79	x(1,15,2)	2680	x(5,5,1)	97.14
y3	149.71	x(2,5,2)	4231.67	NODES	71
y4	144.83	x(3,1,2)	4900.59	CPU time (s)	28.116s
y5	190.74	x(4,9,2)	5052.97		
y6	220.61	x(5,3,2)	2158.51		
y7	156.54	x(5,7,2)	2600		
y8	168.2	x(6,3,2)	2524.48		
y9	182.95	x(6,10,2)	2087.52		
y10	180.48	x(7,8,2)	3291.69		
y11	174.42	x(7,19,2)	1500		
y12	142.57	x(8,4,2)	3476.29		
y13	202.88	x(8,20,2)	1000		
y14	146.88	x(9,10,2)	3824.96		
y15	119.72	x(9,20,2)	1900		

Table 39: Solution of s15-p15-c15-seg3

x(1,8,3)	4466.54	y1	166.97
x(2,9,3)	5042.12	y2	146.77
x(3,11,3)	4738.52	y3	171.6
x(4,5,3)	4639.94	y4	160.59
x(5,3,3)	4780.35	y5	156.27
x(6,7,3)	4393.85	y6	192.91
x(7,3,3)	4201.1	y7	217.48
x(8,1,3)	5085.51	y8	206.59
x(9,15,3)	4528.89	y9	184.96
x(10,1,3)	4534.91	y10	162.85
x(11,6,3)	3000.01	y11	167.26
x(11,10,3)	3000.01	y12	164.4
x(12,13,3)	5179.68	y13	150.69
x(13,3,1)	109.46	y14	161.32
x(13,6,1)	248.58	y15	128.91
x(13,14,3)	3866.67	NODES	779
x(14,7,1)	137.04	CPU time (s)	1054.261
x(14,11,3)	5440.99		
x(15,10,3)	4948.86		

Table 40: Solution to s15-p15-c20-seg2

y1	160.62	x(1,3,2)	4306.65	x(10,4,2)	4560	x(19,8,2)	2193.96
y2	142.3	x(2,11,2)	4269.87	x(10,12,1)	24.18	x(20,9,2)	2051.43
y3	167.18	x(3,1,2)	2000.01	x(11,5,2)	4521.77	x(20,10,2)	2799.99
y4	158.28	x(3,12,2)	2514.11	x(12,6,1)	489.46	Nodes	1981
y5	153.83	x(4,8,2)	2518.12	x(12,13,2)	3811.3	CPU time (s)	2973.754
y6	190.27	x(4,10,2)	2000.01	x(13,7,2)	4549.54		
y7	215.22	x(5,6,2)	4759.23	x(14,15,2)	4196.35		
y8	210.03	x(6,9,2)	4548.15	x(15,5,2)	4358.21		
y9	179.28	x(7,3,1)	661.6	x(16,11,2)	2435.02		
y10	163.49	x(7,7,1)	416.82	x(16,14,2)	2000.01		
y11	163.49	x(7,14,2)	3840	x(17,2,2)	3086.67		
y12	165.71	x(8,2,2)	2675.56	x(17,15,2)	2307.31		
y13	144.79	x(8,9,1)	58.31	x(18,1,2)	2347.68		
y14	159.52	x(8,13,2)	2191.53	x(18,6,2)	2880.92		
y15	125.6	x(9,1,2)	5471.28	x(19,3,2)	2560.38		

REFERENCES

Abdel-Malek, L.L., Areeratchakul, N., A quadratic programming approach to the multi-product newsvendor problem with side constraints, European Journal of Operational Research, 2007, 176, 1607-1619

Abdel-Malek, L.L., Montanari, R., Morales, L.C., Exact, Approximate, and Generic Iterative Models for the Multi-product Newsboy Problem with Budget Constraint, International Journal of Production Economics, 2004, 91, 189-198

Abdel-Malek, L.L., Montanari, R., An analysis of the multi-product newsboy problem with a budget constraint, International Journal of Production Economics, 2005, 296-307

Abdel-Malek, L.L., Montanari, R., On the multi-product newsboy problem with two constraints, Computers and Operations Research, 2005, 32, 2095-2116

Achterberg, T., Koch, T., Martin, A., Branching rules revisited, Operations Research Letters, 2005, 33, 42 – 54

Ahmed, S., King, A.J., Parija, G., A Multi-Stage Stochastic Integer Programming Approach for Capacity Expansion under Uncertainty, Journal of Global Optimization, 2003, 26, 3-24

Areeratchakul, N., Abdel-Malek, L., An approach for Solving the Multi-product Newsboy Problem, 2006, Vol. 3, No.3, 219-227

Albornoz, V. M., Benario, P., Rojas, M. E., A two-stage stochastic integer programming model for a thermal power system expansion, International Transactions in Operations Research, 2004, 11, 243-257

Alonso-Ayuso, A., Escudero, L.F., Garin, A., Oriuno, M.T., Perez, G., An Approach for Strategic Supply Chain Planning under Uncertainty based on Stochastic 0-1 Programming, Journal of Global Optimization, 2003, 26, 97–124

Bienstock, D., Computational Study of a Family of Mixed Integer Quadratic Programming Problems, Mathematical Programming, 1996, 2, 121-140

Bonami Pierre, Jon Lee, 2006, BONMIN Users' Manual

Bonami, P., Biegler, L.T., Conn, A.R., Cornuejols, G., Grossmann, I.E., Laird, C.D., Lee, J., Lodi, A., Margot, F., Sawaya, N., wachter, A., An algorithm Framework for Convex Mixed Integer Nonlinear Programs, 2005, RC23771, W0511-023

Borchers, B. (MINLP) Branch and Bound Methods, Encyclopedia of Optimization, vol. 3, Kluwer, 2001, pp. 331–335.

Dakin, R.J., "A tree search algorithm for mixed integer programming problems," Computer Journal, 1965, vol. 8, pp. 250–255

Erlebacher, S.J. Optimal and Heuristic Solutions for the Multi-item Newsvendor Problem with A single Capacity Constraint, Production and Operations Management, 2000, 9, 3, ABI/INFORM Global pg. 303

Fiacco, A.V., Mccormick, G.P., Nonlinear Programming, Sequential Unconstrained Minimization Techniques, John Wiley, New York, 1968

Fourer, R., Gay, D.M., Kernighan, B.W., 2003, AMPL: A Modeling Language for Mathematical Programming, Second edition, Brooks/Cole Thomson Learning

Gay, D.M., Writing .nl files, 2005, Sandia National Laboratories, Technical Report No. 2005-7907P

Geunes, J.P., Ramasesh, R.V., Hayya, J.C., Adapting the newsvendor model for infinite horizon inventory systems, Journal of Production Economics, 2001, 72, 237-250

Gupta, A., Maranas, C.D., McDonald, C.M., Mid-term supply chain planning under demand uncertainty: customer demand satisfaction and inventory management, Computers and Chemical Engineering, 2000, 24, 2613–2621

Gupta, A., Maranas, C.D., A Two-Stage Modeling and Solution Framework for Multisite Midterm Planning under Demand Uncertainty, Industrial & Engineering Chemistry Research, 2000, 39, 3799-3813

Gupta, A., Maranas, C.D., Managing demand uncertainty in supply chain planning, Computers and Chemical Engineering, 2003, 27, 1219-1227

Gupta, O.K. and Ravindran,V., Branch and bound experiments in convex nonlinear integer programming, Management Science 31 (1985), no. 12, 1533–1546.

Haneveld, W.K.K., Vlerk, M.H.V.D., Stochastic integer programming: General models and algorithms, Annals of Operations Research, 1999, 85, 39-57

Jaffe, D., Muirhead, T., Tey, T., Avutu, T., Implementation Study: Dell IT Scales Supply Chain Management with Oracle RAC 10g, www.dell.com

Kalvelagen, E., Some MINLP Solution Algorithms, http://www.gams.com/~erwin/minlp

Kim, B., Leung, J.M.Y., Taepark, K., Zhang, G., Lee, S., Configuring a manufacturing firm's supply network with multiple suppliers, 2002, 34, 663-677

Ko, H.J., Evans, G.W., A genetic algorithm-based heuristic for the dynamic integrated forward/reverse logistics network for 3PLs, Computers and Operations Research, 2007, 34, 346-366

Ladanyi, L., Lee, J., Lougee-heimer, R., Rapid Prototyping of Optimization Algorithms Using COIN-OR: A Case Study Involving the Cutting-Stock Problem, Annals of Operations Research, 2005, 139, 243–265

Lau, H., Lau, H., The multi-product multi-constraint newsboy problem: Applications, formulation and solution, Journal of Operations Management, 1995, 13, 153-162

Lau, H., Lau, H., The newsstand problem: A capacitated multiple-product single-period inventory problem, European Journal of Operational Research, 1996, 94, 29-42

Land, A.H. and Doig, A.G., An automatic method for solving discrete programming problems, Econometrica, 1960, vol. 28, pp. 497–520.

Leung, S. C. H., Wu, Y., A two-stage stochastic programming with recourse model for cross-border distribution with fleet management, Production Planning & Control, 2005, Vol. 16, No. 1, 60-70

Leyffer, S., Integrating SQP and branch-and-bound for mixed integer nonlinear programming, Computational Optimization and Applications 18 (2001), 295–309.

Lieckens, K., Vandaele, N., Reverse logistics network design with stochastic lead times, Computers and Operations Research, 2007, 34, 395-416

Lougee-Heimer, R., The Common Optimization Interface for Operations Research: Promoting Open- Source Software in the Operations Research Community. IBM Journal of Research and Development, 2002, 47(1)

Lucas, C., MirHassani, SA, Mitra, G., Poojari, CA, An application of Lagrangian relaxation to a capacity planning problem under uncertainty, Journal of the Operational Research Society, 2001, 52, 1256–1266

Khouja, M., Mehrez, A., A Multi-product Constrained Newsboy Problem with Progressive Multiple Discounts, Computers Industrial Engineering, 1996, Vol. 30, No. 1, 95-101

Nahmias, S., 1989, Production and Operations Analysis, Homewood, IL: Richard D. Irwin.

Niederhoff, J.A., Using separable programming to solve the multi-product multiple ex-ante constraint newsvendor problem and extensions, European Journal of Operational Research, 2007, 176, 941–955

Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P., Numerical Recipes in C: The Art of Scientific Computing, 2nd Edition, Cambridge University Press, 1992, 134-145

Rekik, Y., Sahin, E., Dallery, Y., A comprehensive analysis of the Newsvendor model with unreliable supply, OR Spectrum, 2007, 29, 207–233

Santoso, T., Ahmed, S., Goetschalckx, M., Shapiro, A., A stochastic programming approach for supply chain network design under uncertainty, European Journal of Operational Research, 2005, 167, 96–115

Torabi, S.A., Fatemi Ghomi, S.M.T., Karimi, B., A hybrid genetic algorithm for the finite horizon economic lot and delivery scheduling in supply chains, European Journal of Operations Research, 2006, 173, 173-189

Vairaktarakis, G.L., Robust multi-item newsboy models with a budget constraint, Journal of Production Economics, 2000, 66, 213-226

Wächter, A. and Biegler, L.T., On the Implementation of a Primal-Dual Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming, Mathematical Programming 106(1), pp. 25-57, 2006

Wang, S., Sarker, B.R., A single-stage supply chain system controlled by kanban under just-in-time philosophy, Journal of the Operational Research Society, 2004, 55, 485–494

Wang, S., Sarker, B.R., Optimal models for a multi-stage supply chain system controlled by kanban under just-in-time philosophy, European Journal of Operations Research, 2006, 172, 179-200

Wellons, H. S., Reklaitis, G.V., The design of multiproduct batch plants under uncertainty with staged expansion. Computers and Chemical Engineering, 1989, 13, 115

Weng, Z.K., Coordinating order quantities between the manufacturer and the buyer: A generalized newsvendor model, European Journal of Operational Research, 2004, 156, 148–161

Winston, W.L., Operations Research Applications and Algorithms (Fourth Edition), Brooks/Cole---Thomson Learning, Belmont, CA, US, 2004, 680-688

Yang, S., Yang, J., Abdel-Malek, L.L., Sourcing with random yields and stochastic demand: A newsvendor approach, Computers and Operations Research, 2007, 34, 3682 – 3690

Zhang, G., To appear INFORM, 2007, Combining Acquisition Planning with Inventory Management Under Uncertain Demand

Zhang, G., Ma, L., Optimal Acquisition Policy for a Supply Network with Discount Schemes and Uncertain Demands, International Journal of Production Research, 2007, accepted to publish,

http://www.ampl.com

http://www.gams.com

http://www.opensource.org

http://www.coin-or.org

http://www.coin-or.org/Bonmin

http://www.netlib.org

VITA AUCTORIS

Sicheng Chen was born in 1981 in Haiyang, Shandong, China. He graduated from the East China University of Science and Technology, Shanghai, China in 2004 and obtained a Bachelor of Engineering degree in Chemical Process Automatic Control. He then pursued his studies in Industrial and Manufacturing Systems Engineering at the University of Windsor, Windsor, Ontario, Canada from September 2005. He is currently a Master's degree candidate and intends to pursue his Ph.D studies from January 2008 at McMaster University, Hamilton, Ontario, Canada.