JOINT PRICING AND INVENTORY DECISIONS FOR SUBSTITUTABLE AND PERISHABLE PRODUCTS

by

Aylin Polat

A Thesis Submitted to the
Graduate School of Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Industrial Engineering

Koç University

August, 2013

Koç University
Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Aylin Polat

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

Assist. Prof. Onur Kaya

Assoc. Prof. Lerzan Örmeci

Assist. Prof. Bora Çekyay

Date:

To my family with gratitude

ABSTRACT

This study investigates the problem of jointly determining the order size and optimal prices for a perishable inventory system under the condition that demand is time and price dependent. The inventory is also assumed to decay at a certain rate. We assume that a decision-maker has the opportunity to adjust prices during the sales season to influence demand and to improve revenues. In this system the seller is allowed to change the prices for a discrete number of times at a certain cost. We develop a mathematical model to find the optimal times to change the prices, the optimal prices and the order quantity. We analyze both single and double product system where items are procured from a single supplier with respect to the economic order quantity model. This research investigates the operational efficiency of the supply chain by considering the effects of price change throughout the timeline. We develop analytical results for the optimal prices when the times of price changes are given. However, due to the complexity of the problem, we design heuristic algorithms in order to find best times to change the prices in order to obtain the highest profit values. We develop two different heuristics which are Simulated Annealing and Genetic Algorithm. We observe that better results can be obtained by genetic algorithm for most of the parameter settings. For the double product system, we analyze multiple pricing at equal time intervals and obtain similar results as in the single product case. We again apply Simulated Annealing and Genetic Algorithm heuristic methods in order to find the best time arrays and we again observe that genetic algorithm gives better results than simulated annealing. We observe that the profit values can be significantly increased by using multiple prices over the season rather than using a constant price for both single product and double product cases. The system can be managed much more efficiently and the decay rates can be decreased significantly with the application of multiple prices over the season. Through numerical experiments, we also analyze the effects of different parameters on this system and extract managerial insights.

ÖZETÇE

Bu çalışma talep fonksiyonunun zaman ve fiyata bağlı olduğu durumlar için bozunabilen envanter sistemlerinde en uygun fiyat ve sipariş büyüklüğünün belirlenmesini incelemektedir. Yapılan analizler süresince karar verici kişilerin ürün fiyatlarını değiştirerek talepleri etkileyebileceği ve bu şekilde kazancın artırılabileceği varsayılmıştır. Sistemde satıcının önceden belirlenen zaman aralıklarında belirli bir maliyeti üstlenerek ürünlerin fiyatlarını değiştirmesine izin verilmiştir. Fiyatların değiştirilmesi gereken en uygun zaman aralıkları, bu zaman aralıklarında uygulanması gereken fiyatlar ve sipariş büyüklüğünü bulmak üzere matematiksel modeller geliştirilmiştir. Bu çalışma hem tek ürünlü hem de çift ürünlü sistemler için uygulanmış ve yapılan analizler sırasında ürünlerin tek bir tedarikçiden temin edildiği varsayılmıştır. Bu araştırma aynı zamanda belirli zaman çizelgesi boyunca fiyat değişikliğinin etkilerini dikkate alarak tedarik zincirinin operasyonel verimliliğini de incelemektedir. Fiyat değişim sayıları önceden belirlenerek zaman çizelgelerine uygun elde edilebilecek en iyi fiyatları bulmak üzere analitik modeller geliştirilmiştir. Ancak sorunun karmaşıklığı nedeniyle en yüksek kâr değerlerini elde etmek için fiyat değişim zamanları sezgisel algoritmalar tasarlanarak elde edilmiştir. İki farklı sezgisel algoritma kullanılmıştır bu çalışmada: genetik algoritma ve benzetimli tavlama algoritması. Genetik algoritma ile çok daha iyi sonuçlar elde edildiği görülmüştür. İki ürünlü sistemlerde ise eş zaman aralıklarında fiyat değişikliği yapıldığı varsayılarak aynı şekilde sezgisel algoritma çalışmaları yapılmış ve tek ürünlü sistemdekine benzer olarak genetik algoritma ile daha iyi sonuçlar elde edilmiştir. Tüm yapılan çalışmaların sonucunda hem tek ürünlü hem de iki ürünlü sistemlerde tek fiyat yerine birden fazla fiyat uygulaması yapıldığı takdirde daha yüksek kâr değerlerine ulaşıldığı görülmüştür. Bu sistemlerde bozunma oranlarında azalma ve aynı zamanda operasyonel verimlilikte artış sağlanmıştır. Bu sonuçlara ek olarak farklı parametre etkileri analiz edilerek yönetsel çıkarımlar yapılmıştır.

ACKNOWLEDGMENTS

I first wish to express my sincere gratitude to my supervisor, Dr.Onur Kaya. His invaluable insight and experience, guidance, and encouragement have had a great impact on this thesis. I consider myself to be truly fortunate to have had the opportunity to work with him.

We are thankful to Tubitak for partially supporting this work through grant #111M533. I am extremely grateful to the two readers on my MSc dissertation committee, Dr.Lerzan Örmeci, and Dr.Bora Çekyay.

I wish to thank Beste Günay, who is always more than a friend for me. Her endless care and support helped me to overcome setbacks and stay focused on my graduate study.

My special thanks to Eray Saltik; his love and patience made me to overcome the difficulties which made me feel the strength to finish my thesis. He was always there cheering me up and stood by me through good times and bad. Without him it would be impossible for me to continue to my studies.

On a personal note, I express my gratitude to my parents and my brothers for their lifetime support,encouragement and love.

TABLE OF CONTENTS

List of Tables ix

List of Figures xi

Chapter 1: Introduction 1

Chapter 2: Literature Review 6

Chapter 3: Single Product System 12

3.1 Solution Methodology 25

3.1.1 Genetic Algorithm 25

3.1.2 Simulated Annealing 33

Chapter 4: Two Product System 38

4.1 Pricing at Equal Time Intervals 40

4.1.1 Solution Methodology 49

Chapter 5: Computational Studies 52

5.1 Data Generation 52

5.2 Parameter Settings for the Algorithms 54

5.2.1 Parameter Setting for Genetic Algorithm 54

5.2.2 Parameter Setting for Simulated Annealing 56

5.2.3 Analysis of the Single Product System 56

5.2.4 Analysis of the Double Product System 78

Chapter 6: Conclusions 87

Bibliography 88

Vita 94

LIST OF TABLES

3.1 Parameter Values for Single Product System 13
3.2 Holding Cost Function Parameters 18
3.3 Roulette Wheel Selection Results 31

4.1 Parameter Values for Double Product System 41

5.1 Single Product System Parameters 53
5.2 Double Product System Parameters 53
5.3 Population Size Determination for Single and Double Product System 54
5.4 Selection Methods for Single and Double Product System 55
5.5 Single and Double Product Mutation Probabilities Comparison 55
5.6 Single Product Simulated Annealing Results at Different Temperatures 56
5.7 Single Product Constant Pricing Results for the Base Parameters 57
5.8 Comparisons of optimal pricing strategies regarding N 57
5.9 Numerical Results with f=0 64
5.10 Numerical Results with f=5 64
5.11 Numerical Results with f=50 65
5.12 The Comparison of Profit Values between Equal Interval Time Line Simple Heuristic Solution and GA with Equal Interval Time Line Population 69
5.13 The Comparison of Profit Values for Single Product System 70
5.14 Sensitivity Analysis with Constant Pricing 71
5.15 Sensitivity Analysis Results with Genetic Algorithm 72
5.16 Sensitivity Comparison with Genetic Algorithm and Constant Pricing 73
5.17 The Comparison Ratios of Genetic Algorithm Sensitivity Results with the Base Case 74
5.18 Sensitivity Analysis Results with Simulated Annealing 75

5.19 Sensitivity Comparison with Simulated Annealing and Constant Pricing . . . 76
5.20 Sensitivity Comparison with Genetic Algorithm Having Equally Spaced Time Intervals . . . 77
5.21 The Comparison of Profit Values for Double Product System . . . 80
5.22 Double Product Sensitivity Results with Constant Pricing . . . 81
5.23 Double Product Sensitivity Analysis Results with Genetic Algorithm . . . 82
5.24 Comparison of Genetic Algorithm Results with Constant Pricing Case in Double Product System . . . 83
5.25 The Comparison Ratios of Genetic Algorithm Sensitivity Results with the Base Case . . . 84
5.26 Double Product Sensitivity Analysis Results with Simulated Annealing Algorithm . . . 85
5.27 Comparison of Simulated Annealing Results with Constant Pricing Case in Double Product System . . . 86

LIST OF FIGURES

3.1 Illustration of Inventory Level for Single Product System 13

3.2 Illustration of Genetic Algorithm 26

3.3 Illustration of Chromosomes 28

3.4 Cooling Schedule Graph of Simulated Annealing 37

4.1 Time and Price Array Creation Flow 39

4.2 Illustration of Chromosomes 49

5.1 Profit Change with respect to N 58

5.2 Average Price Change with respect to N 59

5.3 Optimal Price Change with respect to Time, N=4 59

5.4 Order Cost Change with respect to N 60

5.5 Holding Cost Change with respect to N 61

5.6 Total Cost Change Change with respect to N 61

5.7 Total Profit Change with respect to$f$ 63

5.8 Price Change Pattern with Simulated Annealing, Genetic Algorithm and Constant Pricing at $f = 0$, $h = 1$; $t_N = 4$, $N = 30$ 63

5.9 Total Cost Change with respect to f 63

5.10 Average Price Change with respect to f 64

Chapter 1

INTRODUCTION

Recent researches show that implementing good pricing policies results in bigger revenue than reduction of variable costs and increase in sales volume. A study by McKinsey and Company states that: "Pricing right is the fastest and most effective way for managers to increase profits". According to Rafaat [23] traditional pricing and inventory control methods are generally static due to need for wealth of information and data about customer behaviour and its own cost structure as well as information covering competition and market structure. The fast development of Information technology enables us use of data in a desired manner leading to improvement of pricing technologies. Perishable product systems are more complex in nature hence detailed pricing strategies have to be applied by using these pricing methodologies.

The process which prevents an item from being used for its intended original use is defined as decay or deterioration for instance: (*i*) spoilage of materials like food or vegetables; (*ii*) physical depletion like evaporation of liquids or alcohol; (*iii*) decay, as in radioactive substances or loss of potency as in the photographic films or pharmaceutical drugs. Deterioration involves two different concepts the first of which is about materials becoming obsolete at the end of their lifetime and the other one is about deteriorating items throughout their planning horizon. The second concept deals with both constant and continuous decay covering all materials from blood to radioactive substances. In addition deteriorating items can also be classified with respect to their value or utility as a function of time. Fresh products such as fruits are decreasing value material, prescription drugs are constant value material and lastly wines or antiques are increasing value materials.

We can categorize mathematical models of the deteriorating items by using the following scheme: single vs. multiple items, deterministic vs. probabilistic demand, static vs. varying demand, single period vs. multiple periods, purchase vs. production model, quantity discount, no shortage vs. shortage and also constant vs. changing deteriorating rate. Classical inventory models assume replenishment rate as either infinite or constant which are unrealistic when we think about perishable items. Replenishment rate of materials are mostly influenced by stock on hand and also instantaneous demand rate. Producers generally tend to produce more or less with respect to their inventory on hand and also instantaneous demand rate in the market. Then it would be illogical to take stock level constant without considering replenishment effect especially for the material groups like food, vegetable and perfumes decreasing in value under deterioration.(Bhunia et al. [9])

This thesis investigates some inventory management methodologies of substitutable and perishable items under demand uncertainty. At the expiration time perishable items become inappropriate for consumption either partially or completely. Academics and also practitioners continuously seek for ways to improve the management of perishable items. Market researches show that if there are multiple products in the system, customers show willingness to buy fresher product which is generally failed to be incorporated in the mathematical models of the inventory systems. Bret [10] states that there are four major groups for perishable items: grocery industry takes more attention which accounts for over 50.4% of the $400 million retail sales in the US . Additionally, medical area accounts for $\$643 billion$ in 2006, and in the area of blood management 75 million units of blood are donated worldwide every year. Therefore it is very important to employ effective inventory management methods in order to obtain maximum efficiency in the supply chain of perishable products.

The major contributions of this dissertation are two-fold. The first contribution is the development of heuristic models in order to evaluate the expected profit function for managing the inventories of multiple product and single product systems under product substitution and perishability. The second contribution is a comparison of the best order-up-to levels that maximize profit by constant pricing case using heuristic algorithms under product perishability. Inventory problems can be divided into single or multiple product cases; in our study we deal with both single and double product systems. Inventory decisions are based on a single set of parameters for single product systems which includes product, its buyer and supplier. Multiple product case includes also relation between products as multiple product model parameters; for instance demand correlation or product substitution. Demand values can be both stochastic and deterministic; in our study we analyze deterministic case.

In multiproduct systems it is hard for the manufacturers to decide upon the pricing and production of the items, because of the products' substitution property. If we change one of the product's price than another product's demand value also change. Producers generally choose superior product substitute inferior one; hence most of the times extra stock belong to the superior product stay unsold, whereas inferior products encounter stockouts. As a result double product pricing decisions are more complex than the single product system decisions(YinPing [40]).

Most inventory models deal with single item in literature; on the other hand in the real world this occurs rarely. Demand may favor the first item in the presence of second material in the inventory; hence may companies deal with several items in their warehouses in order to get most efficient result. This procedure leads many researches to study multi product systems. As in the single product case the purpose is either maximization of profit or minimization of operational costs in the system. Therefore the analysis for a single item inventory is almost same with double product case. As the inventory of deteriorating item; so sooner they are sold out, better than they are stored(Abdulhakim [5]).

For the multiproduct case, in order to decrease supply chain costs, most of the people make significant efforts by using improved inventory management. On the other hand, a large partition of retailers still lose millions of dollars due to loss sales and excess inventory; hence it is critical for them to coordinate inventory management with dynamic pricing to achieve higher overall profit. Perishable products come in fresh with a retail price at the beginning of the cycle . However when they face deterioration and their expiry date comes closer, the retailer prices them at a lower price, which attracts customers who are more price sensitive; hence higher profits are achieved. As an example a CPU's price is lowered down throughout their short lifetime, whenever a new CPU is introduced to the market. Retailer's profit is maximized by dynamic pricing and also coordinating inventory/pricing decisions. When we integrate dynamic pricing and ordering decisions, we can match better supply and demand. (Liu [34])

Managing products that grows increasingly less valuable over time is an important issue to handle, because when we give answer incorrectly its cost can be quite high. A quite good example for this situation is Wal-Mart which faces significant challenge while managing with perishable inventories. Wal-Mart's portfolio is generally composed of perishable items like food (fresh products, dairy and bakaery), pharmaceuticals, chemicals (house cleaning agents) and cut flowers. In a 2003 survey, overall unsalable cost of consumer packed perishable products at distributors, supermarkets and drug stores is estimated at $2.57 billion, and %22 of these costs over 500 million dollars were due to expiration in only branded segment (Grocery Manufacturers of America, 2004).

Additionally the huge impact is not only important for consumer goods but also for industrial products, military ordnance, and blood. In order to model the system we consider the following issues: Firstly decrease in value of the product, change in both demand for product due to aging and market conditions. Secondly, during lifetime of the product replenishment of the products and pricing decisions can be made more than once. Therefore by gathering all information we design system and use heuristic methods in order to obtain values giving high profits. (Deniz [13])

In this study, not only dealing with perishable items but we also consider discrete pricing case for both single and double product systems in order to improve supply chain performance. Discrete pricing is the use of temporal price discrimination in order to improve inventory and capacity management enabling us to adjust prices at minimal cost to satisfy demand. In our study we analyze an EOQ model considering coordinated pricing and lot sizing decisions. The distributor procures a single or multiple products from an external supplier and sells them on a single market. Price response function includes only demand value for single product system, whereas these functions include also competition between products for double product systems. Both systems subject to variable procurement cost, a fixed ordering cost and also holding cost. Main aim is to maximize the average profit by choosing an optimal lot size and pricing strategy where seller is allowed to vary the selling prices over time horizon. The contribution is analyzing an EOQ model with price changes which are limited with organizational costs linked with the number of price change. Only changing prices for a few times can contribute to the profit in larger scale by balancing benefits and costs of the price change.

Along the time horizon seller has to decide the prices and when to change or fix them. Results will be in the form of optimal cycle length, prices, optimal times and order quantity. In constant pricing case it is easy to find optimal values, on the other hand for single and multi product case we develop heuristic methods in order to obtain explicit results,(Transchel [16]).

We use two pricing strategies: Firstly, we use constant pricing where seller determines optimal selling price that is constant over an infinite planning horizon and secondly we use dynamic pricing where seller varies the selling price over time. If there is no cost of price change, the optimal case is to change the prices continuously. If there is a cost associated with price change, a continuous price change will not be beneficial, as a result a limited number of price change will be profitable. We will give the comparison between the constant and dynamic pricing case also. In order to obtain optimized problem, we use two stage systems; during first stage the number of price changes is optimized and during second stage optimal pricing and purchasing strategy is applied. Dynamic pricing offers a decreased cost of holding cost and increased demand rate by efficient usage of stock on hand, (Transchel [51]).

The organization of the thesis is as follows. In Chapter 2 we share related literature work about perishable products, and their pricing strategies. Then in Chapter 3, we give our single product problem in detail and give some insights on computational complexity.We also explain heuristic algorithms in order to solve the problem for single product system. . Next in Chapter 4, we describe the double product system and give detail about the heuristic algorithms used. In Chapter 5, we give the details about data generation and results of computational studies. Finally in Chapter 6, we give a summary of the thesis and some future research directions.

Chapter 2

LITERATURE REVIEW

Ghare et al. [18] are one of the first researchers who analyze perishable products by including EOQ models with deterministic demand and exponential decay. On the other hand, Covert et al. [12] have developed an EOQ model for the items whose deterioration patterns follow the Weibull distribution.

Nahmias [43] reviews related literature about determining the ordering policies for both fixed life perishable inventory and also products having exponential decay. He includes deterministic and stochastic demands in the models for both single and multiple products and also formulates other approximations for perishable products with fixed life over a finite horizon. He constructs policies of the order up to type based on the total inventory. As a brief application he includes blood bank management in his study. Rafaat [23] also presents a detailed survey about deteriorating items including constant rate and variable rate of perishability. In addition he also analyzes probabilistic and also deterministic inventory models with deterioration with constant and variable demand rate.

Wee [30] has developed a joint pricing and replenishment policy for a Weibull distribution deteriorating items with partial backordering and quantity discounts where profit is maximized by using the optimal inventory holding time and the selling price. Both Rafaat [23] and Wee [30] assumes finite replenishment rate. Giri et al. [26] have served an up to date review of inventory models after Rafaat's [23] survey including both deterministic and stochastic demand results

Abdulhakim [5] has considered multi item inventory system with deteriorating items. He constructs models optimizing the objective function with respect to all possible different types of demands as non-linear functions of inventory levels. He achieved optimal solution by using Pontryagin Maximum Principle for different types of demand rates. Bhattacharya [7] uses two item deteriorating inventory models with a linear stock dependent demand rate in his research. He uses control parameters in order to maintain continuous supply to the inventory and then forms an objective function to be used for the calculation of the net profits and also losses. He puts forward the steady state optimal control problem subjected to the constraints given by the ordinary differential equations for optimizing the objective function values. If control parameters satisfy the equations then objective function value can be determined. Two-product continuous review inventory models have been studied recently by Yadavalli et al. [58], Yadavalli et al. [59].

Abad [2] has considered inventory and pricing problem about perishable products faced by a reseller. The inventory model allows demand to be partially back ordered which is the generalization of the model studied by Rakesh et al. [47]. If the materials are highly perishable resellers use backlogging in order to control costs when customers demand fresh stock. The closest point to our study is that reseller may change the prices in order to obtain the maximum utilization of fresh stocks and to achieve the maximum profit. The solution procedure is first solving a single non linear equation and then if required two non linear equations. Rakesh et al. [47] explain the effects of markdown pricing applied to the retail goods disposed to decay. The study also shows the effective results of multi pricing system due to product aging by considering the effects on the ordering intervals and quantities Burwell et al. [11] extend the results of Abad [1] by offering shortages in addition to all unit quantity discounts and demand which is decreasing function of price. Abad [1] gives the retailer ability to change prices and to determine optimal lot sizes without the case of shortage.

Giri et al. [26] study an extended EOQ type perishable product inventory model in which demand rate is dependent on on-hand inventory. He keeps unit item cost and ordering costs constant in the model but takes holding cost as both non linear function of time the item held in stock and also functional form of the on-hand inventory. First type of holding cost is more applicable for green vegetables, fruits and breads and the second type model is for volatile liquids, electronic components and radioactive materials. Bhunia et al. [9] also consider two deterministic deteriorating inventory models which allows shortages. The replenishment rate is dependent on the instantaneous inventory level and also on the demand both of which are increasing functions of time. Benkherouf [6] presents an optimal procedure for the inventory system with shortages in order to find a replenishment Schedule where items deteriorate at a constant rate and demand rates decrease over a known and planning horizon.

Chaudhuri et al. [35] develop an inventory model for a deteriorating item with a price dependent demand rate. He considers rate of deterioration as time proportional and price dependence of demand with power law form. Generally price dependence is taken linear and deterioration as constant hence Chaudhuri et al. [35] give a new insight into the problem. This study contributes to the literature by using perishable product with a constant rate of deterioration and also time proportional decay and demand functions.

Sana [36] deals with an EOQ model for perishable items with price dependent demand and partial backorder. Time proportional decay and occurring shortage only at the beginning of the cycle are the main assumptions for the model. The SFI (Shortage followed by inventory) of replenishment is used. Mishra et al. [37] analyze the price determination for an EOQ model under perfect competition. In order to obtain unit prices marginal revenue and marginal cost are employed. This study gives detailed information about market structure of the economy by using perfect competition case.

Panda et al. [45] consider a single item economic order quantity model where demand is stock dependent. Due to deterioration product's price is changed hence significant increase is obtained in demand value. He also examines whether pre discount affects the net profit value or not other than just applying it during the deterioration process.

Transchel et al. [51] compare dynamic and constant pricing cases by using the economic order quantity model with all unit quantity discount and also price sensitive customer demand. The weakness of the given model is the deterministic and stationary environment based on EOQ assumptions. Different from the research of Transchel et al. [51], Panda et al. [45] work with pre discount effects.

Mishra [37] develops an inventory model for deteriorating items where shortages are allowed and partially backlogged and also Singh [48] develops an EOQ model for deteriorating items with linear demand and variable deterioration rate. The model allows shortages and backlogging where backlogging is taken as variable and dependent on the waiting time of the next replenishment. He takes the objective function as the minimization of the total cost.

P-S You [14] deals with perishable inventory models with price and time dependent demand. The seller has the right to change the prices before the end of the sales season so as to improve sales and revenues. The optimal number of prices, price values and order quantities can be found by the proposed model. Dalfard et al. [52] examine single product system with pricing and inventory model applications. The model aims to decide on the optimal prices, inventory and also production decisions. The constructed model is nonlinear hence Hybrid Genetic Algorithm and Simulated Annealing are used in order to solve the problem. Taguchi experimental design method is also used in order to enhance the performance.

Transchel et al. [16] analyze the dynamic pricing of a single product system where monopolist decision maker determines the pricing points along the time horizon. He gives details about the coordinated decision making of optimal prices and ordering of the inventory by concentrating on the issue that the higher inventory level the more demand level. He achieves that with only a few price change, an unprofitable systems can be made profitable. It is hard to find solutions for dynamic pricing model hence author takes price response functions both linear and exponential in order to get insight about the problem and figures out that: if price is low, then demand is more sensitive to price fluctuations than at higher prices. As a result price variations are lower at the beginning of time cycle than at the end of the cycle.

Pan [56] constructs model with deteriorating items having price sensitive demand. He uses Weibull distribution as in the research of Wee (1999). Additionally seller has the right to change the prices for multiple times with which he can find optimal dynamic prices and order quantity maximizing the profit. Bhowmick et al. [8] study a continuous production model for a deteriorating items where he assumes variable production cycle and allows shortages in the model. He uses different rates of production; hence a large stock of manufactured items at the initial stage can be avoided, which leads to lower holding cost value. DaeSoo et al. [15] study the joint determination of price and lot size; in order to maximize profit which has price dependent demands with non concave objective functions. He uses Kuhn-Tucker conditions by the application of geometric programming.

Gupta et al. [19] show that in a deterministic setting when a reduction in reservation price is applied this leads to declining optimal prices and also he develops heuristic methods in order to obtain near optimal results. Pekelman [20] deals with price determination and production scheduling over a time horizon where demand is time dependent.

Burke et al. [22] study fresh material allocation to shelf space by using heuristic and metaheuristic algorithms. Firstly he studies single product system and develops greedy algorithms, then he extends his solution to the multiproduct case. Bitran et al. [24] research over optimal pricing policies with perishable products having demand correlation; substitution between products is included in the model where customers arrive according to a stochastic process . Drezner et al. [21] analyze the perishable products with respect to no substitution, full substitution and partial substitution by using single product economic order quantity formulation. He proves that full substitution can never be the optimal case, only partial or no substitution may be the optimal cases.

Markdown price is studied by Widyadana et al. [25] where they show that larger profits can be obtained by markdown prices especially for the perishable materials having price dependent demand. Lee et al. [28] are one of the researchers who deal with price discount problems. He extends the results of Monahan by relaxing implicit assumptions of order for order policy adopted by the supplier. Widyadana et al. [25] develop a deteriorating inventory model to increase retailers' profit. Retailer determines the prices in advance and applies markdown price once in a replenishment time. The author puts forward a hypothesis that markdown policy can be used for increasing profit.

Deng et al. [29] analyze the single product pricing decisions by taking in to account a capacity constrained manufacturer having price sensitive demands. He shows the linkage between optimal prices and capacity. Thomas [32] considers a single product system which has deterministic demand. He extends the results of Wagner and Whitin' s [54] inventory work by including price-production decisions. Continuous pricing is not profitable when high amount of price change costs exist in the system. As a result, Netessine [44] analyzes the limited number of price change in a dynamic, deterministic environment where demand depends on current price and time.

Fleischmann et al. [39] contribute to the literature on dynamic pricing by developing deterministic and finite horizon dynamic programming model taking into account price/demand effect as well as a stockpiling/consumption effect. He puts forward endogenous demand and develops analytical decisions into the nature of optimal prices. Whereas Nagare et al. [42] study continuously deteriorating items having random shelf life. An EOQ model is constructed including the zero lead time and constant deterioration rate. Porteus [46] studies with an EOQ model where he includes reduced set up costs and also determines sale rate.

Yinping et al. [40] research multiproduct case with downward substitution. N.Jeyanthi et al. [41] consider the multiproduct system where the main objective is to reduce the holding cost by an effective inventory management. Singh et al. [48] work with the inventory models for deteriorating items having constant demand rates where demand is piecewise linear function; he proposes an inventory replenishment policy. Wee [30] contributes to the literature by using variable demand rate for the single product systems. He formulates an optimal replenishment policy for the items with linear price function of demand.

Most of the researches are about discrete time and fixed points in time where a price change can be applied; or in the case of continuous pricing models allow continuous price change having mostly finite planning horizon. We investigate the dynamic pricing on ordering decisions in classical EOQ context with a given or optimized number of price changes in every cycle for both single and double product system. We construct model by only giving time values random and giving the seller the ability of change the prices so as to maximize the total profit. We also analyze two extreme cases which are constant and continuous pricing. In order to obtain time intervals we put forward heuristic algorithms and their comparisons with constant pricing case.

Chapter 3

SINGLE PRODUCT SYSTEM

In single product system, our main aim is to decide on the pricing and inventory decisions. Inventory cycle is defined as the time period between two successive orders of the new materials along which we try to maximize the profit by pricing at different time intervals. In our model h is the inventory holding cost, N is the number of price changes and f is the cost of price change where Nf gives the cost of total price change. Figure 3.1 is the graphical inventory level demonstration of a single product system.

Orders are replenished in batches of size Q, along every T periods during infinite planning horizon. Each single batch is associated with a procurement cost C per unit. It is assumed that there are no capacity constraints and the orders are delivered with no lead time. Back-orders are not allowed. The retailer determines the number of different prices, N per order cycle length t_N. The menu costs associated with price changes are denoted by $K(N)$ and are non decreasing functions of N. For a given N, the order interval $[0, t_N]$ is partitioned into N mutually exclusive intervals, labelled by $[t_0, t_1), [t_1, t_2), \cdots, [t_{N-1}, t_N)$ where p_i is charged for $[t_{i-1}, t_i)$ interval. t_N is equal to the cycle length T and $t_0 = 0$. Optimal prices are shown by "*".

In our model p_i is the price amount given for the interval $[t_{i-1}, t_i)$. During the same price we obtain decreasing demand value due to demand function's dependency on the time value. In our model inventory decreases due to both decay and demand for the product. We define inventory level as $I(t)$ and Q is the inventory level at the beginning of cycle length. We take decay rate constant given by θ and $w(t) = \theta I(t)$ as the total rate of decay at any time. Parameters are given in Table 3.1.

In Figure 3.1, $x\, axis$ is the time line having N different prices and $y\, axis$ is the inventory level at each time interval. In this figure $t_1, t_2 ..., t_N$ are the times where we apply price adjustments and $I(t_1), I(t_2)..., I(t_N)$ are their corresponding inventory levels. $I(t)$ is the inventory level at time t. $I(t)$ is the amount of inventory at time t which compensates the

Table 3.1: Parameter Values for Single Product System

$I(t)$	net stock at time t for the product, (units).
$I_0 = I(0)$	maximum net stock for the product, (units).
$Q = I_0$	the batch size for the product, (units).
$D(p_i, t)$	demand rate at time t for the product, (units/period).
$\sigma(t)$	wastage coefficient at time t for the product.
p_i	is the price value during $[t_i - t_{i-1}]$.
$\omega(t) = \sigma(t).I(t)$	wastage rate at time t for the product, (units/period).

selling and decay amounts in the following intervals.

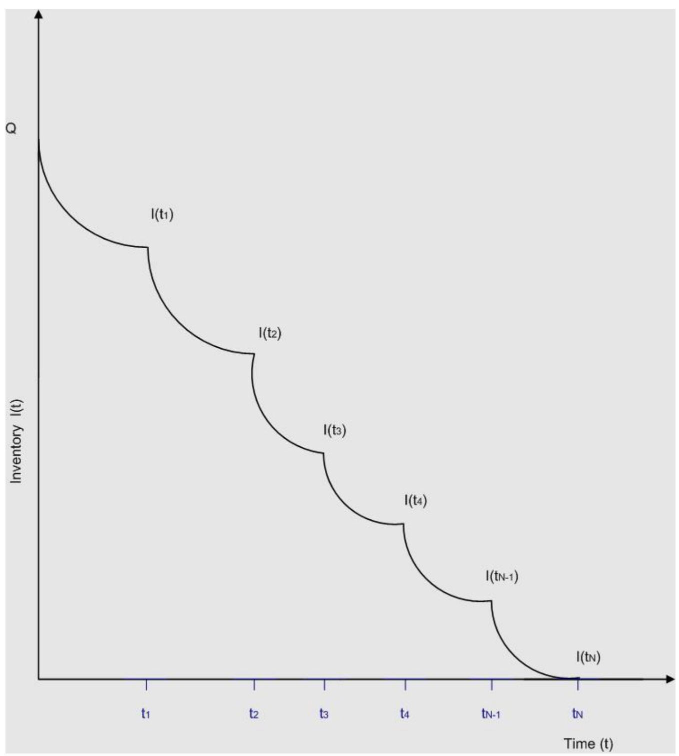


Figure 3.1: Illustration of Inventory Level for Single Product System

For ease of computation backward insertion is done; we start with $I(t_N) = 0$ and use it at the proceeding interval t_{N-1} in order to find $I(t_{N-1})$ and we use $I(t_{N-1})$ to find $I(t_{N-2})$. This solution path is applied for all time intervals. For instance at time t, inventory level is the difference from time t to t_N. At each time interval, demand values are different due to decay and different price.

As a result, inventory computation is done separately at different intervals. In our case, firstly we find the inventory between t and t_2 which gives the inventory decrease due to decay and sell then we sum up with $I(t_2)$. $I(t_2)$ is the inventory covering sell and decay amount from t_2 to t_N. By using backward insertion we can find all $I(t_i)$ up to $I(t_2)$ so we can achieve $I(t)$. $I(t)$ be the instantaneous inventory level at any time $t \geq 0$.

Demand rate $D(p_i, t)$ is assumed to be positive having a negative derivative in its entire domain. The inventory is partly depleted to satisfy demand and partly for deterioration. Along the time line given in Figure 3.1, t_N gives reorder point and for each N different prices, inventory levels can be evaluated at N different points. For $t\epsilon[0, t_N]$, the net stock is positive. For each time interval i boundary value c is the amount of inventory at beginning of the proceeding time interval given by I_{t_i}.

We take demand as $D(p_i, t) = a - \beta p_i - dt$ where p_i gives price amount at i^{th} price adjustment point. During time products perish at a certain amount, hence product's demand decreases with respect to decay amount which is included as $d\,t$ in the demand equation. The instantaneous state of $I(t)$ at any time t is described by the differential equation given by Eq. 3.1.

$$\begin{aligned} I(t) &= \int_t^{t_N} \{D(p_i, t) + w(t)\} dt \\ I(t) &= \int_t^{t_N} \{D(p_i, t) + \sigma(t) I(t)\} dt \end{aligned} \tag{3.1}$$

The deterioration of product is continuous and a constant fraction $(0 < \theta < 1)$ of the on-hand inventory deteriorates per unit time. Deterioration rate θ is deterministic, known and constant. There is no replenishment or repair of deteriorating items during the inventory cycle. We use constant value in order to obtain explicit results, otherwise it gets impossible to reach optimal time, inventory and also profit values.

$$I(t) = \int_t^{t_N} \{D(p_i, t) + \theta I(t)\} dt \tag{3.2}$$

Differentiating Eq. 3.2 with respect to t, we have

$$\frac{\partial I(t)}{\partial t} = -D(p_i, t) - \theta I(t) \tag{3.3}$$

If we write Eq. 3.2 in the form,

$$\frac{\partial I(t)}{\partial t} + \theta I(t) = -D(p_i, t) \tag{3.4}$$

Eq. 3.4 is a first order differential equation with variable coefficients. This equation can be solved by using integrating factor shown by $\mu(t)$. The integrating factor is $\mu(t) = e^{\theta t}$. Multiplying Eq. 3.4 by $\mu(t)$, we obtain

$$\begin{aligned} e^{\theta t}\frac{dI(t)}{dt} + \theta e^{\theta t} I(t) &= -e^{\theta t} D(p_i, t) \\ \frac{d(e^{\theta t} I(t))}{dt} &= -e^{\theta t} D(p_i, t) \end{aligned} \tag{3.5}$$

By integrating both sides of Eq. 3.5 we find that

$$e^{\theta t} I(t) = \int_t^{tN} e^{\theta s} D(p_i, s) ds + c \tag{3.6}$$

where c is an arbitrary constant. Note that we have used s to denote the integration variable to distinguish it from the independent variable. By solving Eq. 3.6 for $I(t)$ we obtain the general solution

$$I(t) = e^{-\theta t} \int_t^{tN} e^{\theta s} D(p_i, s) ds + ce^{-\theta t} \tag{3.7}$$

For any $t \epsilon$ $[t_{i-1}, t_i]$ satisfies the following inventory equations,Eq. 3.3 and also $I_i(t_i) = I(t_i)$.

$$\begin{aligned} I_i(t) &= \int_t^{t_i} (D(p_i, s) + I(s)\theta) ds + I(t_i) \\ I_i(t) &= \int_t^{t_i} D(p_i, s) e^{\theta(s-t)} ds + I(t_i) e^{\theta(t_i - t)} \\ I_i(t) &= \int_t^{t_i} (a - \beta p_i - ds) e^{\theta(s-t)} ds + I(t_i) e^{\theta(t_i - t)} \end{aligned} \tag{3.8}$$

Integral in Eq. 3.8 gives the amount of sell and decay amount of product between t and t_i, $I(t_i)$ is the inventory amount at the beginning of the following interval. As a result at any time t, $I(t)$ is the inventory level at time t. When we integrate the Eq. 3.8

$$I_i(t) = \frac{a-\beta p_i}{\theta}e^{\theta(t_i-t)} - \frac{dt_i}{\theta}e^{\theta(t_i-t)} + \frac{d}{\theta^2}e^{\theta(t_i-t)} - \frac{a-\beta p_i}{\theta} + \frac{dt}{\theta} - \frac{d}{\theta^2} + I(t_i)e^{\theta(t_i-t)} \quad (3.9)$$

The change of inventory at any time t is the result of demand and decay which is given by Eq. 3.3, should hold to verify the Eq. 3.9.

$$\begin{aligned} \frac{dI_i(t)}{dt} &= -(a-\beta p_i - dt) - \theta I_i(t) \\ \frac{dI_i(t)}{dt} &= -(a-\beta p_i)e^{\theta(t_i-t)} + dt_i e^{\theta(t_i-t)} - \frac{d}{\theta}e^{\theta(t_i-t)} + \frac{d}{\theta} - \theta I(t_i)e^{\theta(t_i-t)} \end{aligned} \quad (3.10)$$

When we multiply $I(t)$ with $-\theta$ and subtract $D = a - \beta p_i - dt$

$$\begin{aligned} -(a-\beta p_i - dt) - \theta I(t) = &-(a-\beta p_i)e^{\theta(t_i-t)} + dt_i e^{\theta(t_i-t)} \\ &- \frac{d}{\theta}e^{\theta(t_i-t)} + \frac{d}{\theta} - I(t_i)\theta e^{\theta(t_i-t)} \end{aligned} \quad (3.11)$$

Hence Eq. 3.11 is equal to Eq. 3.10, which verifies that Eq. 3.9 correctly denotes $I_i(t)$. The selling amount of product at i^{th} interval, denoted by S_i, is the total demand belonging to that interval found by Eq. 3.12.

$$\begin{aligned} S_i &= \int_{t_{i-1}}^{t_i} (a - \beta p_i - dt)dt \\ &= a(t_i - t_{i-1}) - \beta p_i(t_i - t_{i-1}) - \frac{d}{2}(t_i^2 - t_{i-1}^2) \end{aligned} \quad (3.12)$$

We find total amount of product sold for the whole inventory cycle by summing up all S_i values for each time interval, given by Eq. 3.13.

$$\begin{aligned} S &= \sum_{i=1}^{N} \int_{t_{i-1}}^{t_i} (a - \beta p_i - dt)dt \\ &= \sum_{i=1}^{N} a(t_i - t_{i-1}) - \beta p_i(t_i - t_{i-1}) - \frac{d}{2}(t_i^2 - t_{i-1}^2) \end{aligned} \quad (3.13)$$

Total inventory Q is the inventory level at time $t = 0$. Hence $I(0)$ gives the initial inventory level for the whole inventory cycle, given by Eq. 3.14 which is also given in Figure 3.1 where it is denoted by Q.

$$I(0) = Q = \frac{a - \beta p_1}{\theta} e^{\theta(t_1 - t_0)} - \frac{dt_1}{\theta} e^{\theta(t_1 - t_0)} + \frac{d}{\theta^2} e^{\theta(t_1 - t_0)} - \frac{a - \beta p_1}{\theta} - \frac{d}{\theta^2} + \frac{dt_0}{\theta} + I(t_1) e^{\theta(t_1 - t_0)} \tag{3.14}$$

In order to define the initial inventory equation explicitly, we write inventory level equations stepwise as given in Eq. 3.15. We need $I(t_1)$ value in order to find I(0) which is given below. Inventory level at t_N; $I(t_N)$ is zero hence by using backward insertion we obtain initial inventory, Q, as given in Eq. 3.16.

$$\begin{aligned} I(t_1)e^{\theta t_1} &= \frac{a - \beta p_2}{\theta} e^{\theta(t_2)} - \frac{dt_2}{\theta} e^{\theta(t_2)} + \frac{d}{\theta^2} e^{\theta(t_2)} \\ &- \frac{a - \beta p_2}{\theta} e^{\theta t_1} - \frac{d}{\theta^2} e^{\theta t_1} + \frac{dt_1}{\theta} e^{\theta t_1} + I(t_2) e^{\theta(t_2)} \\ I(t_2)e^{\theta t_2} &= \frac{a - \beta p_3}{\theta} e^{\theta(t_3)} - \frac{dt_3}{\theta} e^{\theta(t_3)} + \frac{d}{\theta^2} e^{\theta(t_3)} \\ &- \frac{a - \beta p_3}{\theta} e^{\theta t_2} - \frac{d}{\theta^2} e^{\theta t_2} + \frac{dt_2}{\theta} e^{\theta t_2} + I(t_3) e^{\theta(t_3)} \\ &\vdots \\ I(t_{N-1})e^{\theta t_{N-1}} &= \frac{a - \beta p_{t_N}}{\theta} e^{\theta(t_N)} - \frac{dt_N}{\theta} e^{\theta(t_N)} + \frac{d}{\theta^2} e^{\theta(t_N)} \\ &- \frac{a - \beta p_N}{\theta} e^{\theta t_{N-1}} - \frac{d}{\theta^2} e^{\theta t_{N-1}} + \frac{dt_{N-1}}{\theta} e^{\theta t_{N-1}} + I(t_N) e^{\theta(t_N)} \end{aligned} \tag{3.15}$$

$$Q = \sum_{i=1}^{N} \frac{a - \beta p_i}{\theta} e^{\theta(t_i)} - \frac{dt_i}{\theta} e^{\theta(t_i)} + \frac{d}{\theta^2} e^{\theta(t_i)} - \frac{a - \beta p_i}{\theta} e^{\theta(t_{i-1})} - \frac{d}{\theta^2} e^{\theta(t_{i-1})} + \frac{dt_{i-1}}{\theta} e^{\theta(t_{i-1})} \tag{3.16}$$

Inventory equation can also be written in general form given by Eq. 3.17.

$$I(t_j) = \sum_{i=j+1}^{N} \frac{a - \beta p_i}{\theta} e^{\theta(t_i)} - \frac{dt_i}{\theta} e^{\theta(t_i)} + \frac{d}{\theta^2} e^{\theta(t_i)} - \frac{a - \beta p_i}{\theta} e^{\theta(t_{i-1})} - \frac{d}{\theta^2} e^{\theta(t_{i-1})} + \frac{dt_{i-1}}{\theta} e^{\theta(t_{i-1})} \tag{3.17}$$

In a single product system, we find revenue in an interval by multiplication of total demand and price during that demand period. Total revenue is the sum of revenue earned at each interval. In our system total revenue is found as follows,

$$\begin{aligned} Revenue &= \sum_{i=1}^{N} \int_{t_{i-1}}^{t_i} (a - \beta p_i - dt) p_i dt \\ &= \sum_{i=1}^{N} a p_i (t_i - t_{i-1}) - \beta p_i^2 (t_i - t_{i-1}) - \frac{d p_i}{2} (t_i^2 - t_{i-1}^2) \end{aligned} \quad (3.18)$$

In the single product supply chain system, there are mainly two performance criteria: order of frequency and the amount of inventory carried along the supply chain. Because of the economies of scale, we don't want to order too frequently and also carry too much inventory. Here the main approach is focusing on the *long run averages* over time. The system operates forever, over the time interval $[0, \infty)$, and we measure the performance by the following quantities,

$$\overline{I} = average\, inventory = \lim_{T \to \infty} \{ (\frac{1}{T}) \int_0^T I(t) dt \}$$

We can identify the long run average $\overline{I}$ by examining the one cycle, the time interval between the receipts of two successive orders. Also, during a cycle, $I(t)$ decreases from Q to *zero*. We can have economies of scale in the supply process by choosing a large Q, but this leads to large average inventory; we can economize on inventory, but only at the expense of a higher order cost and frequency. We measure all costs in some standard monetary unit; here, we use the term *moneys*. Particularly parameters used in total order cost function is given in Table 3.2.

Table 3.2: Holding Cost Function Parameters

k	fixed cost to place an order(moneys)
C	variable cost to place an order (moneys /quantity-unit)
h	cost to hold one unit in inventory for one unit of time

h is the value that at time t, $I(t)$ causes cost to increase at a rate of $hI(t)$. k represents all order costs independent of order size. It includes administrative order processing costs as well as transportation and receiving costs. Hence k represents economies of scale in the supply process. C includes the unit purchase cost as well as any other costs that do not depend on the order size. The total cost per order is thus $k + CQ$. Holding cost h incurs direct costs associated with inventory itself, including costs for physical handling, insurance, refrigeration, and warehouse rental. We denote all these costs by $\underline{h}$. the second component is a financing cost α where α is an interest rate, reflecting the fact that holding inventory ties up capital. Hence, $h = \underline{h} + \alpha C$. In the long run average order cost is $(k + CQ)/t_N$, and the average inventory-holding cost is $h\overline{I}$. The overall performance is sum of these two quantities,

$$C(Q) = total\, average\, cost\, (moneys/time - unit)$$

$$= (k + CQ)/t_N + h\overline{I}$$

$$order\, of\, frequency = \overline{OF} = \frac{1}{t_N}$$

Then average cost is,

$$C(t_N) = (k + CQ)\overline{OF} + h\overline{I}$$

As a first step, we calculated the total cost for the inventory cycle. For each time interval, we take the area under the inventory curve and then multiply with the unit holding cost to find the total holding cost. By using Eq. 3.9,

$$\begin{gathered} h\overline{I} = holding\, cost = \tfrac{h}{t_N} \sum_{i=1}^{N} \int_{t_i-1}^{t_i} I_i(t)dt \\ = \tfrac{h}{t_N}\Big(\sum_{i=1}^{N} \int_{t_{i-1}}^{t_i} \big(\tfrac{a-\beta p_i}{\theta} e^{\theta(t_i-t)} - \tfrac{dt_i}{\theta} e^{\theta(t_i-t)} \\ + \tfrac{d}{\theta^2} e^{\theta(t_i-t)} - \tfrac{a-\beta P_i}{\theta} + \tfrac{dt}{\theta} - \tfrac{d}{\theta^2} + I(t_i)e^{\theta(t_i-t)} \big) dt \Big) \end{gathered} \tag{3.19}$$

We integrate Eq. 3.19, then holding cost equation becomes

$$\begin{gathered} h\overline{I} = \tfrac{h}{t_N} \sum_{i=1}^{N} \big(- \tfrac{a-\beta p_i}{\theta^2}(1 - e^{\theta(t_i-t_{i-1})}) + \tfrac{dt_i}{\theta^2}(1 - e^{\theta(t_i-t_{i-1})}) \\ - \tfrac{d}{\theta^3}(1 - e^{\theta(t_i-t_{i-1})}) - \tfrac{a-\beta p_i}{\theta}(t_i - t_{i-1}) \\ + \tfrac{d}{2\theta}(t_i^2 - t_{i-1}^2) - \tfrac{d}{\theta^2}(t_i - t_{i-1}) + \tfrac{I(t_i)}{-\theta}(1 - e^{\theta(t_i-t_{i-1})}) \big) \end{gathered} \tag{3.20}$$

Profit is the difference between total revenue and total cost. So profit function can be written as follows, assuming N price changes are made and $I(t_i)$ is included in open form.

$$\pi^{(N)} = \frac{1}{t_N}\sum_{i=1}^{N}\big\{ap_i(t_i - t_{i-1}) - \beta p_i^2(t_i - t_{i-1}) - \frac{dp_i}{2}(t_i^2 - t_{i-1}^2)$$
$$- h[-\frac{a-\beta p_i}{\theta^2}(1 - e^{\theta(t_i - t_{i-1})}) + \frac{dt_i}{\theta^2}(1 - e^{\theta(t_i - t_{i-1})})$$
$$- \frac{d}{\theta^3}(1 - e^{\theta(t_i - t_{i-1}}) - \frac{a-\beta p_i}{\theta}(t_i - t_{i-1})$$
$$+ \frac{d}{2\theta}(t_i^2 - t_{i-1}^2) - \frac{d}{\theta^2}(t_i - t_{i-1}) - \sum_{j=i+1}^{N}\big(\frac{a-\beta p_j}{\theta^2}e^{\theta(t_j)}$$
$$- \frac{dt_j}{\theta^2}e^{\theta(t_j)} + \frac{d}{\theta^3}e^{\theta(t_j)} - \frac{a-\beta p_j}{\theta^2}e^{\theta(t_{j-1})} - \frac{d}{\theta^3}e^{\theta(t_{j-1})}$$
$$+ \frac{dt_{j-1}}{\theta^2}e^{\theta(t_{j-1})}\big)(1 - e^{\theta(t_i - t_{i-1})})] - (k + CQ)\big\} \tag{3.21}$$

Main aim of the problem is to maximize the profit, done by constructing two stages. At the second stage: there is a given number of price adjustments and by using that, optimal timing decisions and also optimal prices are found. Eq. 3.25 gives the average profit for each given number of pricing decisions which includes unit revenue minus direct purchasing, holding and set up cost over the cycle length.

Then different profit values are found with respect to given different number of pricing decisions for a fixed cycle length t_N. Then first stage is used to find the optimal number of pricing, done by calculating profits for different n values and most profit giving one is chosen as the optimal. Here $K(N)$ gives the cost of price change. If you are changing price N times then cost of price change is $K(N)$ which is included in the profit function at the first stage.

$$Stage1: \tag{3.22}$$

$$\pi^* = max\pi^{(N)} - K(N) \tag{3.23}$$

$$Stage2: \tag{3.24}$$

$$\begin{aligned}\pi^{(N)} = max \quad & \frac{1}{t_N}\sum_{i=1}^{N}\{ap_i(t_i - t_{i-1}) - \beta p_i^2(t_i - t_{i-1}) - \frac{dp_i}{2}(t_i^2 - t_{i-1}^2) \\ & - h[-\frac{a-\beta p_i}{\theta^2}(1 - e^{\theta(t_i - t_{i-1})}) + \frac{dt_i}{\theta^2}(1 - e^{\theta(t_i - t_{i-1})}) \\ & - \frac{d}{\theta^3}(1 - e^{\theta(t_i - t_{i-1})}) - \frac{a-\beta p_i}{\theta}(t_i - t_{i-1}) \\ & + \frac{d}{2\theta}(t_i^2 - t_{i-1}^2) - \frac{d}{\theta^2}(t_i - t_{i-1}) - \sum_{j=i+1}^{N}(\frac{a-\beta p_j}{\theta^2}e^{\theta(t_j)} \\ & - \frac{dt_j}{\theta^2}e^{\theta(t_j)} + \frac{d}{\theta^3}e^{\theta(t_j)} - \frac{a-\beta p_j}{\theta^2}e^{\theta(t_{j-1})} - \frac{d}{\theta^3}e^{\theta(t_{j-1})} \\ & + \frac{dt_{j-1}}{\theta^2}e^{\theta(t_{j-1})})(1 - e^{\theta(t_i - t_{i-1})})] - (k + CQ)\}\end{aligned} \tag{3.25}$$

$$s.t \qquad D_i \geq 0 \qquad \forall i \in i = 1, \cdots, N \tag{3.26}$$

$$t_{i-1} - t_i \leq 0 \qquad \forall i \in i = 1, \cdots, N \tag{3.27}$$

$$t_0 \geq 0$$

Eq. 3.26 prevents any negative value for the demand, which is unacceptable in real life cases. Eq. 3.27 guarantees that all time intervals are mutually exclusive and exhaustive.
In order to maximize the average profit, we differentiate Eq. 3.25 with respect to p_i and t_i for $i = 1, \ldots, N$. The necessary first order conditions for the optimal prices $\frac{\partial \pi^N}{\partial p_i} \overset{!}{=} 0$ and the optimal times for price changes $\frac{\partial \pi^N}{\partial t_i} \overset{!}{=} 0$ for $i = 1, \ldots, N$ and $j = 1, \ldots, N$.
In order to find optimal prices for the given t_i values, we differentiate Eq. 3.25 with respect to p_i for $i = 1, \ldots, N$ and $i = 1, \ldots, N$. If we are dealing with the i^{th}interval, for the intervals other than $t_i - t_{i-1}$ the differentiation with respect to p_i results in zero. For instance in order

to find p_1^* we only deal with the interval $t_1 - t_0$ and use the Eq. 3.28 for the differentiation.

$$\begin{aligned}
\pi_1^{(1)} &= ap_1(t_1 - t_0) - \beta p_1^2(t_1 - t_0) - \frac{dp_1}{2}(t_1^2 - t_0^2) \\
&- h\big[- \frac{a - \beta p_1}{\theta^2}(1 - e^{\theta(t_1 - t_0)}) + \frac{dt_1}{\theta^2}(1 - e^{\theta(t_1 - t_0)}) \\
&- \frac{d}{\theta^3}(1 - e^{\theta(t_1 - t_0}) - \frac{a - \beta p_1}{\theta}(t_1 - t_0) + \frac{d}{2\theta}(t_1^2 - t_0^2) \\
&- \frac{d}{\theta^2}(t_1 - t_0) - \sum_{j=2}^{N} \big(\frac{a - \beta p_j}{\theta^2} e^{\theta(t_j)} - \frac{dt_j}{\theta^2} e^{\theta(t_j)} \\
&+ \frac{d}{\theta^3} e^{\theta(t_j)} - \frac{a - \beta p_j}{\theta^2} e^{\theta(t_{j-1})} - \frac{d}{\theta^3} e^{\theta(t_{j-1})} \\
&+ \frac{dt_{j-1}}{\theta^2} e^{\theta(t_{j-1})} \big)(1 - e^{\theta(t_1 - t_0)}) \big] - (k + CQ)
\end{aligned} \tag{3.28}$$

After we differentiate Eq. 3.28 with respect to p_1 and the result is given in Eq. 3.29.

$$p_1^* = \frac{a}{2\beta} - \frac{d}{4\beta}(t_1 + t_0) - \frac{h}{2\theta} + \frac{(e^{\theta(t_1 - t_0)} - 1)}{\theta(t_1 - t_0)} \frac{h}{2\theta} + \frac{C(e^{\theta(t_1)} - e^{\theta(t_0)})}{2\theta(t_1 - t_0)} \tag{3.29}$$

Theorem 3.1 *For a given set of t_i values, the optimal prices p_i^* are found using the result of first order derivative of profit function given in Eq. 3.30. Profit takes its maximum value at the given p_i^*. π is a function which is twice differentiable at p_i and satisfies $\frac{\partial^2 \pi}{\partial p_i^2} < 0$. Hence π has global maximum point at p_i, which satisfies both Eq .(3.31) and Eq .(3.32).*

$$p_i^* = \frac{a}{2\beta} - \frac{d}{4\beta}(t_i + t_{i-1}) - \frac{h}{2\theta} + \frac{(e^{\theta(t_i - t_{i-1})} - 1)}{\theta(t_i - t_{i-1})} \frac{h}{2\theta} + \frac{C(e^{\theta(t_i)} - e^{\theta(t_{i-1})})}{2\theta(t_i - t_{i-1})} \tag{3.30}$$

$$\begin{aligned}
\frac{\partial \pi}{\partial p_1} &= a(t_1 - t_0) - 2\beta p_1(t_1 - t_0) - \frac{d}{2}(t_1^2 - t_0^2) \\
&- h\big[\frac{\beta}{\theta^2}(1 - e^{\theta(t_1 - t_0)} + \frac{\beta}{\theta}(t_1 - t_0)\big] - \big(C\frac{\partial Q}{\partial p_1}\big)
\end{aligned} \tag{3.31}$$

$$\frac{\partial Q}{\partial p_1} = \frac{-\beta}{\theta}(e^{\theta(t_1)} - e^{\theta(t_0)})$$

$$\frac{\partial^2 \pi}{\partial p_1^2} = -2\beta(t_1 - t_0) \tag{3.32}$$

In order to find t_i^*, we both deal with the intervals $(t_i - t_{i-1})\, and\, (t_i - t_{i+1})$. When we take derivative of the profit function with respect to t_i, the values other than the given intervals

$(t_i - t_{i-1})\, and\, (t_i - t_{i+1})$ become zero. For instance, t_1* can be found by using the Eq. 3.34.

$$\begin{aligned}
\pi^{(2)} &= ap_1(t_1 - t_0) - \beta p_1^2(t_1 - t_0) - \frac{dp_1}{2}(t_1^2 - t_0^2) \\
&- h\big[- \frac{a - \beta p_1}{\theta^2}(1 - e^{\theta(t_1 - t_0)}) + \frac{dt_1}{\theta^2}(1 - e^{\theta(t_1 - t_0)}) \\
&- \frac{d}{\theta^3}(1 - e^{\theta(t_1 - t_0})) - \frac{a - \beta t_1}{\theta}(t_1 - t_0) + \frac{d}{2\theta}(t_1^2 - t_0^2) \\
&+ \frac{I(t_1)}{-\theta}(1 - e^{\theta(t_1 - t_0)}) + [- \frac{a - \beta p_2}{\theta^2}(1 - e^{\theta(t_2 - t_1)}) \\
&+ \frac{dt_2}{\theta^2}(1 - e^{\theta(t_2 - t_1)}) - \frac{d}{\theta^3}(1 - e^{\theta(t_2 - t_1}) - \frac{a - \beta t_2}{\theta}(t_2 - t_1) \\
&+ \frac{d}{2\theta}(t_2^2 - t_1^2) + \frac{I(t_2)}{-\theta}(1 - e^{\theta(t_2 - t_1)})\big] - (k + CQ)
\end{aligned} \tag{3.33}$$

We can write $I(t_1)$ and $I(t_2)$ explicitly in Eq. 3.34 as given below.

$$\begin{aligned}
\pi^{(2)} &= ap_1(t_1 - t_0) - \beta p_1^2(t_1 - t_0) - \frac{dp_1}{2}(t_1^2 - t_0^2) \\
&+ ap_2(t_2 - t_1) - \beta p_2^2(t_2 - t_1) - \frac{dp_2(t_2^2 - t_1^2)}{2} \\
&- h\big[- \frac{a - \beta p_1}{\theta^2}(1 - e^{\theta(t_1 - t_0)}) + \frac{dt_1}{\theta^2}(1 - e^{\theta(t_1 - t_0)}) \\
&- \frac{d}{\theta^3}(1 - e^{\theta(t_1 - t_0}) - \frac{a - \beta p_1}{\theta}(t_1 - t_0) + \frac{d}{2\theta}(t_1^2 - t_0^2) - \frac{d(t_1 - t_0)}{\theta^2} \\
&- \sum_{j=2}^{N} \big(\frac{a - \beta p_j}{\theta^2} e^{\theta(t_j)} - \frac{dt_j}{\theta^2} e^{\theta(t_j)} \\
&+ \frac{d}{\theta^3} e^{\theta(t_j)} - \frac{a - \beta p_j}{\theta^2} e^{\theta(t_{j-1})} - \frac{d}{\theta^3} e^{\theta(t_{j-1})} \\
&+ \frac{dt_{j-1}}{\theta^2} e^{\theta(t_{j-1})}\big)(1 - e^{\theta(t_1 - t_0)})\big] + h[- \frac{a - \beta p_2}{\theta^2}(1 - e^{\theta(t_2 - t_1)}) \\
&+ \frac{dt_2}{\theta^2}(1 - e^{\theta(t_2 - t_1)}) - \frac{d}{\theta^3}(1 - e^{\theta(t_2 - t_1}) - \frac{a - \beta t_2}{\theta}(t_2 - t_1) \\
&+ \frac{d}{2\theta}(t_2^2 - t_1^2) - \frac{d(t_2 - t_1)}{\theta^2} - \sum_{j=3}^{N} \big(\frac{a - \beta p_j}{\theta^2} e^{\theta(t_j)} - \frac{dt_j}{\theta^2} e^{\theta(t_j)} \\
&+ \frac{d}{\theta^3} e^{\theta(t_j)} - \frac{a - \beta p_j}{\theta^2} e^{\theta(t_{j-1})} - \frac{d}{\theta^3} e^{\theta(t_{j-1})} \\
&+ \frac{dt_{j-1}}{\theta^2} e^{\theta(t_{j-1})}\big)(1 - e^{\theta(t_2 - t_1)})\big] - (k + CQ)
\end{aligned} \tag{3.34}$$

$$\begin{aligned}
\frac{\partial I(t_1)}{\partial t_1} &= -(a-\beta p_2)e^{\theta t_1} + dt_1 e^{\theta t_1} \\
\frac{\partial I(t_2)}{\partial t_1} &= 0 \\
\frac{\partial Q}{\partial t_1} &= e^{\theta t_1}(2a-\beta p_1-\beta p_2)
\end{aligned} \tag{3.35}$$

After we differentiate the Eq. 3.34 with respect to t_i, we obtain Eq. 3.36.

$$\begin{aligned}
&\frac{\partial \pi^{(2)}}{\partial t_1} = ap_1 - \beta p_1^2 - dp_1 t_1 - ap_2 + bp_2^2 + dp_2 t_1 \\
&- h\big[\frac{a-\beta p_1}{\theta}e^{\theta(t_1-t_0)} + \frac{d}{\theta^2} - \frac{d}{\theta^2}e^{\theta(t_1-t_0)} - \frac{dt_1}{\theta}e^{\theta(t_1-t_0)} \\
&+ \frac{de^{\theta(t_1-t_0)}}{\theta^2} - \frac{a-\beta p_1}{\theta} + \frac{dt_1}{\theta} - \frac{d}{\theta^2} - (\frac{a-\beta p_2 e^{\theta(t_1)}}{\theta} - \frac{de^{\theta(t_1)}}{\theta^2} \\
&+ \frac{de^{\theta(t_1)}}{\theta^2} + \frac{dt_1 e^{\theta(t_1)}}{\theta})(1-e^{\theta(t_1-t_0)}) + I(t_1)\theta e^{\theta(t_1-t_0)}\big] \\
&+ h[-\frac{a-\beta p_2}{\theta}e^{\theta(t_2-t_1)} - \frac{dt_2}{\theta^2}e^{\theta(t_2-t_1)}(\theta) - \frac{d}{\theta^2}e^{\theta(t_2-t_1)} \\
&+ \frac{a-\beta p_2}{\theta} - \frac{d}{\theta}t_1 + \frac{d}{\theta^2} + I(t_2)\theta e^{\theta(t_2-t_1)}\big] - C[e^{\theta t_1}(2a-\beta p_1-\beta p_2)]
\end{aligned} \tag{3.36}$$

By using Eq. 3.36, we obtained optimal time equation for the first interval which is t_1^* given in Eq. 3.37. t_1^* is the value making the Eq. 3.37 zero.

$$\begin{aligned}
&ap_1 - \beta p_1^2 - dp_1 t_1 - ap_2 + bp_2^2 + dp_2 t_1 \\
&- h\big[\frac{a-\beta p_1}{\theta}e^{\theta(t_1-t_0)} + \frac{d}{\theta^2} - \frac{d}{\theta^2}e^{\theta(t_1-t_0)} - \frac{dt_1}{\theta}e^{\theta(t_1-t_0)} \\
&+ \frac{de^{\theta(t_1-t_0)}}{\theta^2} - \frac{a-\beta p_1}{\theta} + \frac{dt_1}{\theta} - \frac{d}{\theta^2} - (\frac{a-\beta p_2 e^{\theta(t_1)}}{\theta} - \frac{de^{\theta(t_1)}}{\theta^2} \\
&+ \frac{de^{\theta(t_1)}}{\theta^2} + \frac{dt_1 e^{\theta(t_1)}}{\theta})(1-e^{\theta(t_1-t_0)}) + I(t_1)\theta e^{\theta(t_1-t_0)}\big] \\
&+ h[-\frac{a-\beta p_2}{\theta}e^{\theta(t_2-t_1)} - \frac{dt_2}{\theta^2}e^{\theta(t_2-t_1)}(\theta) - \frac{d}{\theta^2}e^{\theta(t_2-t_1)} \\
&+ \frac{a-\beta p_2}{\theta} - \frac{d}{\theta}t_1 + \frac{d}{\theta^2} + I(t_2)\theta e^{\theta(t_2-t_1)}\big] - C[e^{\theta t_1}(2a-\beta p_1-\beta p_2)] = 0
\end{aligned} \tag{3.37}$$

In general form optimal times given by t_i^* can be found by using Eq. 3.38 with the procedure given for the t_1^* value.

$$
\begin{aligned}
&ap_i - \beta p_i^2 - dp_i t_i - ap_{i+1} + p_{i+1}^2 + dp_{i+1}t_1 \\
&- h\big[\frac{a-\beta p_i}{\theta}e^{\theta(t_i - t_{i-1})} + \frac{d}{\theta^2} - \frac{d}{\theta^2}e^{\theta(t_i - t_{i-1})} - \frac{dt_i}{\theta}e^{\theta(t_i - t_{i-1})} \\
&+ \frac{de^{\theta(t_i - t_{i-1})}}{\theta^2} - \frac{a-\beta p_i}{\theta} + \frac{dt_i}{\theta} - \frac{d}{\theta^2} - (\frac{a-\beta p_{i+1}e^{\theta(t_i)}}{\theta} - \frac{de^{\theta(t_i)}}{\theta^2} \\
&+ \frac{de^{\theta(t_i)}}{\theta^2} + \frac{dt_i e^{\theta(t_i)}}{\theta})(1 - e^{\theta(t_i - t_{i-1})}) + I(t_i)\theta e^{\theta(t_i - t_{i-1})}\big] \\
&+ h[-\frac{a-\beta p_{i+1}}{\theta}e^{\theta(t_{i+1}-t_i)} - \frac{dt_{i+1}}{\theta^2}e^{\theta(t_{i+1}-t_i)}(\theta) - \frac{d}{\theta^2}e^{\theta(t_{i+1}-t_i)} \\
&+ \frac{a-\beta p_{i+1}}{\theta} - \frac{d}{\theta}t_i + \frac{d}{\theta^2} + I(t_{i+1})\theta e^{\theta(t_{i+1}-t_i)}\big] - C[e^{\theta t_i}(2a - \beta p_i - \beta p_{i+1})] = 0 \qquad (3.38)
\end{aligned}
$$

In order to find optimal t_i, the Eq. 3.38 has to be solved. Optimal time depends on instantaneous inventory and also on the price values all which are functions of time where exponential and non linear terms prevent us finding explicit solutions. Hence use of heuristic algorithms is reasonable given in Section 3.1.

3.1 *Solution Methodology*

In this section we apply heuristic algorithms in order to find maximum profit giving time arrays; we propose genetic algorithm and also simulated annealing. Optimal price, inventory and profit functions depend on time, hence we obtain all other unknown values by using time arrays generated by heuristic algorithms.

3.1.1 Genetic Algorithm

Overview

Genetic algorithm is an optimization method based on naturally inspired genetic operations. It uses selection, mutation and crossover operations to achieve its optimization goal. Genetic Algorithm is an adaptive strategy and a Global Optimization technique. It is an Evolutionary Algorithm and belongs to the broader study of Evolutionary Computation. The Genetic Algorithm is inspired by population genetics (including hereditary and gene frequencies), and evolution at the population level, as well as the Mendelian understanding of the structure (such as chromosomes, genes, alleles) and mechanisms (such as recombination and mutation).

Individuals of a population contribute to their genetic material proportional to their suitability of their expressed genome (called phenotype) to their environment, in the form of offspring. The next generation is created through a process of mating which involves recombination of two individual genomes in the population with the introduction of random copying errors (called mutation). This iterative procedure may result in improved fit between phenotypes of individuals in a population and the environment.

An implementation of Genetic Algorithm starts with a population of (typically random) chromosomes. One then evaluates these structures and allocate reproductive opportunities in such a way that those chromosomes which represent a better solution to the target problem are given more chance to reproduce than other chromosomes, which are poorer solutions. After parent selection crossover is applied by using one point variable length crossover. In order to have diversification mutation is applied. We form mating pool with the offspring and the original population, then by generating random numbers chromosomes are selected into the next generation having higher fitness value. Until a termination criterion is met this procedure is repeated and the best profit giving price array and time array are selected. The flow of genetic algorithm is given in Figure 3.2.

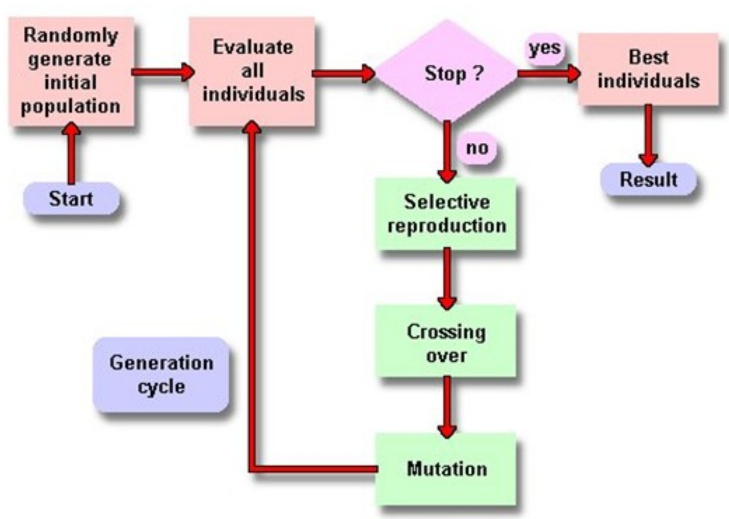


Figure 3.2: Illustration of Genetic Algorithm

Pseudocode for Genetic Algorithm

Classical approach is proposed for genetic algorithm. The outline can be summarized as follows:

Step 1. Generate a random population of chromosomes which are constructed from feasible solutions of time arrays formed by random numbers

Step 2. Evaluate fitness values of chromosomes by using profit functions with the corresponding price functions.

Step 3. Select two chromosomes from the population according to their fitness value (bigger fitness higher probability of selection)

Step 4. According to crossover probabilities, by crossover operator, form new offspring from parents.

Step 5. According to the mutation probabilities, mutate new offspring at each randomly selected locus

Step 6. Place new offspring in the population

Step 7. Select the next population according to their fitness values.

Step 8. Use new generated population for the further run of the algorithm

Step 9. If the end condition is satisfied; code stops and returns the best solution in the population

Representation of the Chromosomes

Genetic Algorithm forms a population and operates on it containing some encoding of the parameters set. In order to encode each solution to a chromosome one string is used for time arrays composed of real numbers. Typically a population is formed by 30-100 individuals. There are types of chromosome representations like permutation, binary and tree encoding, whereas for our problem most suitable one is value encoding. In value encoding each chromosome is sequence of some real values. For totally N number of price change, we use value string having size $(1 * N)$. Chromosome A shows the time sequence at which price adjustments are done by having reorder point as 10. For this chromosome totally six prices are found by using optimal price functions. In our problem reorder points are taken as floating, hence in the population time arrays can be found having size from $(1 * 1)$ to $(1 * N)$.

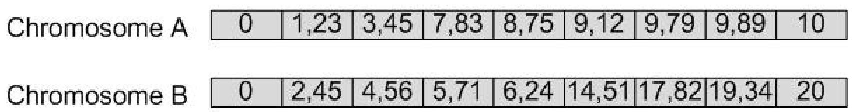


Figure 3.3: Illustration of Chromosomes

Initial Solution Generation

As an initial population, we generate population having chromosomes for only time determination; all other values like price and profit are evaluated by functions. Each individual chromosome in the population is a candidate solution to the problem. We use complete random method to generate initial populations. For each N number time change, we generate value strings randomly with the amount of population size and find optimal prices by using the optimal time functions derived from the objective function. Time strings are the genotype of the problem. Phenotype is the objective function value giving profits which is also taken as the fitness value of the chromosomes. Chromosomes having high fitness values are selected as parents and given the opportunity to reproduce by crossover. Important to note that crossover provides intensification and mutation provides diversification. We try population size of 30, 50 and 100 with genetic algorithm and find that the results are close to each other. Hence we decided using population size 30 is enough through the genetic algorithm for single product system.

Here the main important item to decide is population size because it changes both initial population and also other items like mating pool, offspring and parent selection. Hence in order to decide on the best population size, we try different values on data sets. After deciding on the best population size parameter, we apply other GA tests with the best population size parameter value. It is important that while deciding on the population size, we keep all other parameters same.

Selection

Reproduction is usually the first operator applied on population. From the population, chromosomes are selected as parents in order to apply crossover and produce offspring.

According to the evolution theory survival of the fittest is the major issue to consider, which means that the best ones should survive and create the new offspring. The reproduction operators are also called selection operators. Selection means extract a subset of genes from an existing population according to their fitness values. Fitness quantifies the optimality of a chromosome so that a particular solution may be ranked against all other solutions. The most commonly use methods are Roulette Wheel, Rank, Steady State, Boltzmann, Tournament.

In selection each individual will be an offspring of the two genomes from the previous generation. Most of the time this method works quite well, but sometimes it loses the best individuals to crossover or mutation operation. In order to make genetic algorithm robust, best members of the generation must be kept between generations. Number of best members to be kept unchanged to the next generation is determined by the number of elite variable. After initial solution is generated, it is sorted with respect to fitness values. Before crossover and mutation are applied best chromosomes are kept unchanged by using number of elite which has user defined value.

There are different alternatives for the selection as stated before. In tournament selection procedure, the algorithm chooses two solutions from the mating pool randomly and puts better one into the next population. Here randomly selected ones create diversification in the solution. Firstly two random numbers are generated and by looking these two numbers, the total values under these randomly generated numbers are checked to see which profit value is higher. The next generation is formed by using chromosomes giving higher objective function value.

Sometimes genetic algorithms tend to produce same or similar result by being stuck in local maxima (minima), because gene pool becomes similar so there is no new or different genome to produce different solution. One solution to this issue is introducing new individuals to the gene pool at each generation. But adding new individuals to new gene pool is not enough because if they cannot mate, new genomes cannot be introduced to the next generation. In order to make keep gene pool diversified, roulette wheel selection mechanism is used. Detailed explanation of the roulette wheel selection will be given in the next section. Number of new individuals are kept proportional to size of the population to make the algorithm scalable and defined by the variable new individual.

In deterministic sampling; chromosomes are sorted according to their objective function values and best valued chromosomes are assigned as new generation and they are also initial solution of the next iteration. For the tournament selection, a pair of the chromosome from mating pool is selected randomly for comparison, and the one which has the best objective value is accepted as a parent of new generation(next initial solution).

Roulette Wheel Selection

Roulette wheel selection is one of the reproduction mechanism used in genetic algorithm. Key difference of the roulette wheel selection is it makes the reproduction more randomized because every genome will have a chance to reproduce proportional to its fitness value. Because of its randomized nature, poorly performing genomes which have important information inside could still reproduce and therefore pass their crucial difference to the next generation.

Roulette wheel selection performs badly on negative fitness values. In order to circumvent this situation, fitness values of the genomes are shifted by the amount of minimum fitness value in current generation, making fitness values of all genomes above zero, which gives good probability scaling for reproduction. In the Roulette wheel selection method, the first step is to calculate the cumulative fitness of the whole population through the sum of the fitness of all individuals. After that, the probability of selection is calculated for each individual as being wheel = fitness/total fitness. Then, an array is built containing cumulative probabilities of the individuals. So, N random numbers are generated in the range 0 to wheel and for each random number an array element which can have higher value is searched for. Therefore, individuals are selected according to their probabilities of selection.

Illustration of Selection

Evolutionary algorithms is to maximize the profit function f(x) with x in the real time string [0, n], i.e., x=0, 1.25, 2.75, 3.48, 5 The first step is encoding chromosomes; use value string representation for real value numbers up to predefined reorder point n. Assume population size is 4. Generate initial population at random. They are chromosomes or genotypes; e.g.

x_1 =0, 1.25, 2.75, 3.48, 5

Table 3.3: Roulette Wheel Selection Results

String no	Initial Population	Fitness	Probability	Expected Count
1	0, 1.25, 2.75, 3.48, 5	7463.3	0.288	0.99
2	0, 2.34, 3.85, 4	7513.3	0.249	0.997
3	0, 3.41, 4.56, 8.74, 10	7551.2	0.251	1.002
4	0, 2.66, 7	7615.6	0.252	1.011
sum		30143.6	1	4
Average		7535.85	0.25	1
Max		7615.6	0.253	1.011

x_2 = 0, 2.34, 3.85, 4

x_3 =0, 3.41, 4.56, 8.74, 10

x_4=0, 2.66, 7

Calculate fitness value for each individual

f (x_1)=7463.3, f(x_2)=7513.3, f(x_3)=7551.2, f(x_4)=7615.6

Select parents for crossover based on their fitness in wheel. In roulette wheel the probability of the i^{th}string in the population is $p_i = f_i(\sum_{j=1}^{n} f_j)$, where

f_j is fitness for the string i in the population

p_i is probability of the string being selected

n is the number of individuals in the population

$n * p_i$ is the expected count

As we analyze the Table 3.3, the number of expected count is maximum with chromosome; hence,chromosome 4 selection probability would be higher.

Mutation

After crossover is performed, mutation takes place. Mutation is one of the most crucial parts of the genetic algorithm; it makes the population diverse by changing small parts of the genomes. In this project, mutation is made to change only one point of the genome. In single product system, every genome is mutated by a predefined probability of 0.07 defined by mutation probability variable. Main reason behind this number is most of the genetic algorithm programs uses this number and it turns out to be a good ratio of keeping the population diverse enough to produce better results

Mutation alters one or more gene values in chromosome from its initial state. This can be result in entirely new values being added to the gene pool. With the new gene values, the genetic algorithm may be able to arrive at better solution than was previously possible. Mutation is intended to prevent the search falling into a local optimum of the state space. The mutation operators are Flip Bit, Boundary, Non Uniform, and Gaussian. In our problem Uniform mutation operator is used. In our problem uniform mutation is used. Consider two chromosomes are selected for mutation,

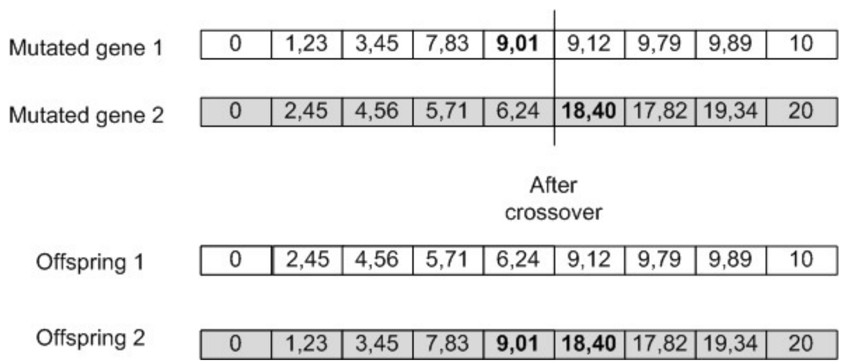
Offspring 1	0	1,23	3,45	7,83	8,75	9,12	9,79	9,89	10
Offspring 2	0	2,45	4,56	5,71	6,24	14,51	17,82	19,34	20

Random numbers are generated where time values are changed into new ones. The mutated chromosomes are as follows

Mutated gene 1	0	1,23	3,45	7,83	**9,01**	9,12	9,79	9,89	10
Mutated gene 2	0	2,45	4,56	5,71	6,24	**18,40**	17,82	19,34	20

Crossover

Crossover is exchange of genetic material to produce offspring in genetic algorithm. Using crossover also means reproduction in which parent selection and recombination are done simultaneously. Crossover mainly takes two parents and cuts them at random position and then swap them. Crossover probability is generally taken between 0.6-1. Crossover can be one point; two-point, uniform or other crossover types can also be used compatible with the problem nature. Crossover methods are differ greatly and could change the performance of the program. Main difficulty is to map the problem into a genetic structure which reflects nature of the problem. In this simulation one-point variable length crossover is used. The main reason behind this motive is by using variable length genomes; genetic algorithm could exploit this feature to search for a solution with different N values which greatly reduced computation cost. If constant genome length was used, programmer has to simulate all possible (or desired) N values in order to have a healthy comparison of performance. In order to apply crossover two chromosomes are selected randomly from the population excluding the chromosomes having largest fitness value by using the number of elite hence loss of good alleles is prevented. Chromosomes are cut at the randomly selected points and then gene exchange is maintained. For each chromosome pair, probability of crossover is randomly generated in each time. The pairs are selected randomly from a 30 member population. Therefore, each iteration fifteen pairs are created and crossover is applied if default crossover probability is less than the random number generated by code as crossover probability. One point crossover operator randomly selects crossover point and then copies everything before this point from the second parent and then everything after the crossover point copy from the first parent. The crossover would then look as shown below. If we consider the two parents selected for crossover, after interchanging the parents' chromosomes before the crossover points, the offspring produced would be as follows,



3.1.2 Simulated Annealing

Overview

This heuristic optimization method is inspired by mostly the thermodynamic properties of cooling matters. In thermodynamics, every particle tries to make their internal energy as low as possible, but in the cooling process, some particles increase their temperature by some probability. Main principle of SA algorithm is same as the thermodynamic laws. A solution tries to minimize (maximize) its output by changing its values. In this simulation, the annealing process tries to maximize its output. While doing this, changes resulting in output increase are accepted automatically. However, output decreasing changes are also accepted by some probability which is inverse proportional to the output difference and proportional to the temperature of the system.

In SA Algorithm initial temperature, cooling schedule, starting solution and neighbourhood structure are the important points that we have to consider. Initial temperature must be high enough to allow free exchange of neighbouring solutions. Cooling schedule determines the rate of convergence and probability of accepting non improving solutions. In order to search adequately in fewer amounts of iterations, neighbourhood should be small, but also it has to be large enough to obtain drastic profit improvements.

In Simulated Annealing, during iterations solutions are found and these values are compared with respect to their objective function values. Improving solutions are always accepted but a fraction of non improving solutions are also accepted. The possibility of accepting non improving solutions depends on the temperature parameter. Simulated annealing allows hill climbing moves worsening objective function value, hence increases the chance of obtaining global optimum. It differs from traditional local search by employing objective function worsening moves. So convergence to local optima is prevented. Temperature value decreases at each iteration, so hill climbing occurs less frequently and makes probable converging to a global optimal solution. Descent strategy can be both steepest descent and also random descent.

Pseudo Code for Simulated Annealing

Input:$Problem_{size}, iteration_{max}, temp_{max}$

Output:S_{best}

$S_{current} \leftarrow$ Create Initial solution ($Problem_{size}$);

$S_{best} \leftarrow S_{current}$;

for i=1 to $iteration_{max}$ do

$S_i \leftarrow$ Create Neighbourhood ($S_{current}$) ;

$Temp_{current} \leftarrow$ Calculate Temperature (i, $temp_{max}$);

if $Cost(S_i) \leq$ Cost ($S_{current}$) then

$S_{current} \leq S_i$

if Cost(S_i) $\leq$ Cost(S_{best}) then

$S_{best} \leq S_i$

end

else if $exp(\frac{Cost(S_{current})-Cost(S_i)}{temp_{current}}) >$ Rand () then ;

$S_{current} \leftarrow S_i$;

end

end

Return S_{best};

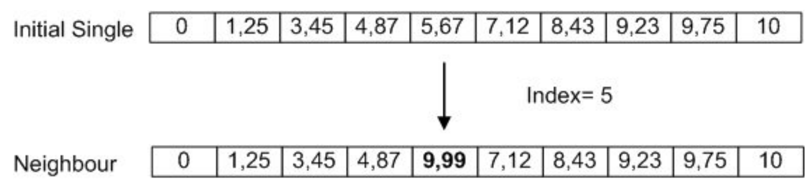
Solution and Neighbour Representations

In the problem we take the number of price change and cycle time as variable parameters and for each value different time arrays are generated by randomization method. If the problem has as input of 7 as the number of price change and 10 as the cycle time, the initial solution will be in the form of $1x9$ array like given below. In every turn of simulated annealing, a

Initial Single	0	1,25	3,45	4,87	5,67	7,12	8,43	9,23	9,75	10

neighbour state is selected and decided to be used or not. There are 3 generally used methods of neighbour selection. Those methods are linear perturbation, Gaussian perturbation and randomization. Perturbation is changing a value to a new value which depends on current value of the system. Perturbation can be described as noise insertion to the system. The value stays more or less the same, but it can change system behaviour. The difference between linear perturbation and Gaussian perturbation is the type of the noise introduced. In linear perturbation, the change is selected from a uniform random distribution, whereas in Gaussian perturbation the noise is selected from a Gaussian distribution. Randomization is changing a point in the solution to a new random one and it is the method used in the simulation. The reason behind this decision is randomization outperforms both Uniform and Gaussian perturbation in terms of profit value. If the solution stuck on a local maxima, perturbation methods came insufficient to move the solution to a new solution region and that's the main reason randomization outperforms perturbation.

Initial solution is composed of time series having randomly generated values. In order to create neighbour for our current solution, a random number is generated in order to be used as an index number in which price change is done. In order to have a neighbour the move operator finds an index and changes its value by a different random number. Below random number is generated as 5; move operator changes the value in index 5 by 9.99.



Temperature

If the profit value of the newly generated time series is larger than the current value then solution is directly accepted as the neighbour of our solution. If the result is worse than the current one, it still has the chance of being a neighbour which is the most remarkable point that SA differs from ordinary scatter searches.

Acceptance value is evaluated in the system by using the objective function values of the current and new time series given as follows: acceptance=exp (new solution-current solution)/temperature. It can be appreciated that as the temperature of the system decreases the probability of accepting a worse move is decreased. This is the same as gradually moving to a frozen state in physical annealing.

Also note, that if the temperature is zero then only better moves will be accepted which effectively makes simulated annealing act like hill climbing. Temperature can be described as how much bad change can I tolerate at this moment. A high temperature system is unstable; various changes in output which results worse output can occur, gradually the temperature drops and the system becomes stable and behaves like a greedy algorithm. Temperature decreases by the principle of exponential decay. In the simulation, temperature is defined by the variable temp and it is selected different on each simulation.

Initial temperature selection is also made after trying several temperatures as 2,10,30 and 100.Initial temperature should be high enough to allow a move to almost all neighbourhoods. If it is not so, final solution may be so close to initial solution. It means initial temperature should allow a hill climbing moves. However, if the initial temperature starts too high values, search may move disorderly and transform into a random search.The cooling schedule graph is given in Figure 3.4 whose value changes according to the exponential temperature function given below where Temp is temperature.

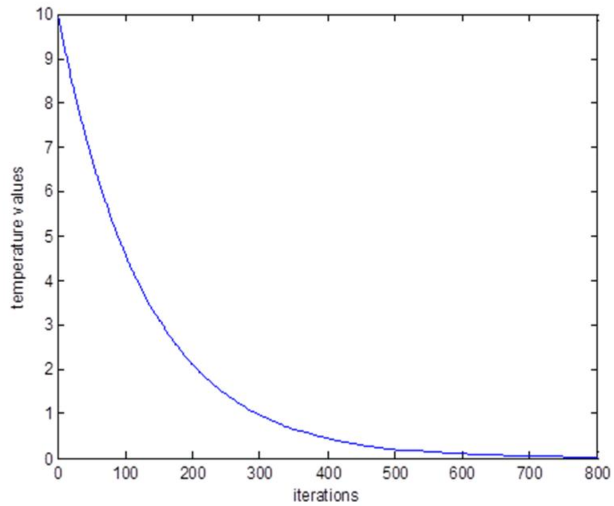


Figure 3.4: Cooling Schedule Graph of Simulated Annealing

In single product system, temperature is set to 2 because of trivial nature of the problem; a small temperature leads to more aggressive approach and forces the algorithm to look for better solutions constantly without sacrificing its current position. Such simulations show greedy behaviour in solving of the problem.

Pseudocode for Simulated Annealing

Step 1.: Generate time arrays randomly for both single and double product system taking cycle time n as the end point and "0" as the starting point. These arrays shows the intervals during which price adjustments are done.

Step 2. Evaluate optimal price values By using the time intervals. These price functions are the found by using the first derivative of profit values.

Step 3. Evaluate holding costs, revenues, fixed and variable costs for each time interval

Step 4. Find index numbers and assign real time values to those preceding indexes randomly

Step 5. Evaluate profit value of the time array. If profit value is smaller than the previous solution's than it is accepted as the neighbour solution.

Step 6.Repeat procedure until termination criteria is met

Chapter 4

TWO PRODUCT SYSTEM

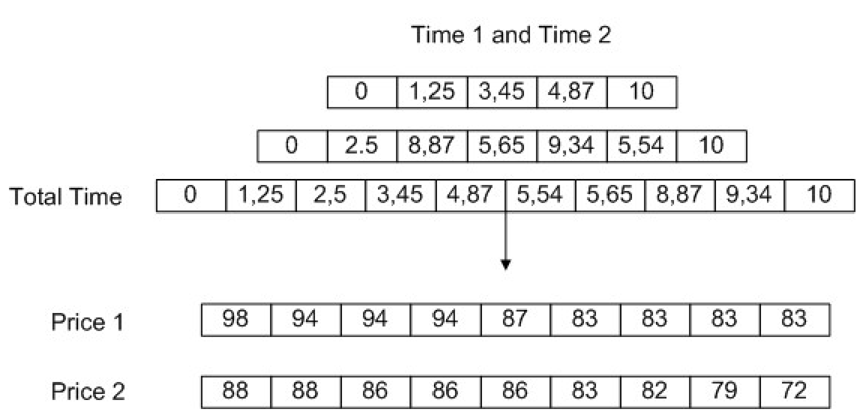
We analyze perishable two product system where products are substitutable. Each product's demand is affected from each other's due to substitution property when these products' value and also freshness change. The problem formulation is very similar with the single product case. We assume replenishment cycle same for both of the products, which is also a very close assumption to the real life case where generally similar products are supplied from a single supplier and orders are given during the same replenishment cycles. Due to high fix costs it would be also illogical to order at different cycle times. On the other hand, in order to obtain higher profit and use the perishable products' inventory more efficiently, we can change their prices at multiple points. We denote times as t_i^1, t_i^2 and prices as p_i, q_i for the first and second product respectively. During each time interval prices are changed but change for which product is unknown. As a result, it is hard to write the profit function without knowing where t_i^1 lies with respect to t_i^2 values. Because without this data, it will be wrong when we try decide on the demand values for each product during time intervals. Hence all t_i^1 and t_i^2 values are taken into single time array denoted by t_i containing all p_i and q_i in order with respect to the given time array. When we construct the price arrays matching with the time values, there is an important point; while searching for each time value in the first or second product's time set, price should be taken constant if the searched time value could not be found in the related product's time set. We use same procedure with single product case in order to find profit, price and inventory values as given in previous section. Details are given below.

In two product system, we assume demand values of the products depend on each other. Evaluation of revenue and cost is same with the single product system. In this system,we take both linear and exponential price responses. The important thing is substitution between the products and we take this into account by analysing the interaction between the prices. In the linear case, when substitute product's price is increased then demand for the first product increases. In the formulation this is provided by using opposite price signs. Deterioration is again used in the formulation by using time factor. Linear Price response for two product system

$$D^1(p_i, q_i, t) = a_1 - \beta_1 p_i + c_1 q_i - d_1 t_i \qquad D^2(p_i, q_i, t) = a_2 - \beta_2 q_i + c_2 p_i - d_2 t_i$$

formation of the two product pricing system: pricing at equal time intervals and also pricing at different time intervals. When we deal with pricing at different time intervals, we form price and time arrays different from the pricing at equal time intervals.For instance lets assume t_i^1 and t_i^2 are the given sets of time arrays which are taken into one single array whose starting and end points are 0 and t_N respectively. Length of the time array is the sum of t_i^1 and t_i^2 and given by N. In the Figure 4.1 visualize the idea of price and time array formation. If we assume all the price and time arrays are given like below, first step is to form T array composed of first and second product's time array, hence we can combines all values in a single size. Time 1 and Time 2 are the values for t_i^1 and t_i^2 and $t_N = 10$. Along time line T we search for the intervals whether the they belong to the Time 1 or Time 2 array. Such that, in this example first interval is $[0 - 1.25]$, we look for the value in the sets and find that it belongs to Time 1 set. Hence first product's price value is changed from 98 to 94, then look for the second interval $[1.25 - 2.5]$ which is in the Time 2 set. As a result first product's price is kept same with the preceding interval which is 94 and second product's price is changed from 88 to 86.

Figure 4.1: Time and Price Array Creation Flow



All the price sets are formed according to this general idea. By keeping in mind, in this system we assume all the price sets and time values are given. We can not find any optimal time or price values as in the other models. Demand value of each product affects the other one as stated before, hence without knowing the exact time periods it is hard to deal with the application of pricing at different time intervals. When we try to apply local maximum theory by taking derivative of profit function with respect to both price and time, no explicit result can be found without knowing the time values. There can be any possibility for a product in a single time interval; its price may change or stay constant both of the conditions affect profit function in a different way. On the other hand, when we adjust prices at the same periods for both of the products, we can get optimal price values by using the results of first derivative of the profit function knowing that profit has a local maximum point. Time values are found by heuristic methods with the same procedure given in single product system. Hence in this thesis, we deal with double product system with price adjustments at equal time intervals. Details are given below.

4.1 *Pricing at Equal Time Intervals*

In two product system, if products are substitute and we price them at the same time intervals, then inventory level graph behaves as if there exists a single product in the system.

Let $I^1(t)$ and $I^2(t)$ be the instantaneous inventory level at any time $t \geq 0$ for the first and second product. Demand rates for the products $D^1(p_i, q_i, t)$ and $D^2(p_i, q_i, t)$ are assumed to be positive having a negative derivative in its entire domain. The inventory is partly depleted to satisfy demand and partly for deterioration. For $t\epsilon[0, t_N]$, the net stock is positive.

Inventory Level Equations

As a first step, we derive the instantaneous state of $I^1(t)$. The procedure is the same for the second product, hence we don't give detailed information for the second product formulation. The general solution for $I^1(t)$ is given in Eq. 4.1.

$$I^1(t) = e^{-\theta_1 t} \int_t^{tN} e^{\theta_1 s} D^1(p_i, q_i, s) ds + ce^{-\theta_1 t} \tag{4.1}$$

Table 4.1: Parameter Values for Double Product System

$I^1(t)$	net stock at time t for the first product, (units).
$I_0^1 = I^1(0)$	maximum net stock for the first product, (units).
$Q^1 = I_0^1$	the batch size for the first product, (units).
$D^1(p_i, q_i, t)$	demand rate at time t for the first product, (units/period).
$\sigma_1(t)$	wastage coefficient at time t for the first product.
$\omega_1(t) = \sigma_1(t).I^1(t)$	wastage rate at time t for the first product, (units/period).
$I^2(t)$	net stock at time t for the second product, (units).
$I_0^2 = I^2(0)$	maximum net stock for the second product, (units).
$Q^2 = I_0^2$	the batch size for the second product, (units).
$D^2(p_i, q_i, t)$	demand rate at time t for the second product, (units/period).
$\sigma_2(t)$	wastage coefficient at time t for the second product.
$\omega_2(t) = \sigma_2(t).I^2(t)$	wastage rate at time t for the second product, (units/period)

Along the time line, t_N gives reorder point and for each N different price adjustments, inventory levels can be evaluated at N different points.For each time interval i boundary value c is the amount of inventory at beginning of the proceeding time interval given by I_i.

We take demand as $D(p_i, q_i, t) = a_1 - \beta_1 p_i + c_1 q_i - d_1 t$ where p_i gives price amount at i^{th} price adjustment point for the first product and q_i gives the price of the second product. During time products perish at a certain amount, hence product's demand decreases with respect to decay amount which is included as $d_1 t$ in the demand equation and also demand increases when there is a price increase in the second product due to their substitution property given by $c_1 q_i$. When we use demand in open form Eq. 4.1 takes the form

$$I(t) = \int_t^{t_i} (a_1 - \beta_1 p_i + c_1 q_i - d_1 s) e^{\theta_1 (s-t)} ds + I^1(t_i) e^{\theta_1 (t_i - t)} \tag{4.2}$$

Integral in Eq. 4.2 gives the amount of sell and decay amount of product between t and t_i, $I^1(t_i)$ is the inventory amount at the beginning of the following interval. When we integrate the Eq. 4.2, we obtain general inventory level equation for both single and double product given in Eq. 4.3 and Eq. 4.4.

$$\begin{aligned} I_i^1(t) = & \frac{a_1 - \beta_1 p_i + c_1 q_i}{\theta_1} e^{\theta_1 (t_i - t)} - \frac{d_1 t_i}{\theta_1} e^{\theta_1 (t_i - t)} + \\ & + \frac{d_1}{\theta_1^2} e^{\theta_1 (t_i - t)} - \frac{a - \beta_1 p_i + c_1 q_i}{\theta_1} + \frac{d_1 t}{\theta_1} - \frac{d_1}{\theta_1^2} + I^1(t_i) e^{\theta_1 (t_i - t)} \end{aligned} \tag{4.3}$$

$$I_i^2(t) = \frac{a_2 - \beta_2 q_i + c_2 p_i}{\theta_2} e^{\theta_2(t_i - t)} - \frac{d_2 t_i}{\theta_2} e^{(t_i - t)} + \\ + \frac{d_2}{\theta_2^2} e^{\theta_2(t_i - t)} - \frac{a_2 - \beta_2 q_i + c_2 p_i}{\theta_2} + \frac{d_2 t}{\theta_2} - \frac{d_2}{\theta_2^2} + I^2(t_i) e^{\theta_2(t_i - t)} \tag{4.4}$$

The selling amounts of the first product and the second product at i^{th} interval, denoted by S_i^1 and S_i^2, is the total demand belonging to that interval found by Eq. 4.5 and Eq. 4.6.

$$S_i^1 = \int_{t_{i-1}}^{t_i} (a - \beta p_i + c_1 q_i - d_1 t) dt \\ = a_1(t_i - t_{i-1}) - \beta_1 p_i (t_i - t_{i-1}) - \frac{d_1}{2}(t_i^2 - t_{i-1}^2) \tag{4.5}$$

$$S_i^2 = \int_{t_{i-1}}^{t_i} (a_2 - \beta_2 q_i + c_2 p_i - d_2 t) dt \\ = a_2(t_i - t_{i-1}) - \beta_2 q_i (t_i - t_{i-1}) + c_2 p_i (t_i - t_{i-1}) - \frac{d_2}{2}(t_i^2 - t_{i-1}^2) \tag{4.6}$$

We find total amount of first product sold for the whole inventory cycle by summing up all S_i^1 values for each time interval, given by Eq. 4.7 and for the second product we find by summing up all S_i^2 values given by Eq. 4.8

$$S^1 = \sum_{i=1}^{N} \int_{t_{i-1}}^{t_i} (a_1 - \beta_1 p_i + c_1 q_i - d_1 t) dt \\ = \sum_{i=1}^{N} a_1(t_i - t_{i-1}) - \beta_1 p_i (t_i - t_{i-1}) + \\ + c_1 q_i (t_i - t_{i-1}) - \frac{d_1}{2}(t_i^2 - t_{i-1}^2) \tag{4.7}$$

$$S^2 = \sum_{i=1}^{N} \int_{t_{i-1}}^{t_i} (a_2 - \beta_2 q_i + c_2 p_i - d_2 t) dt \\ = \sum_{i=1}^{N} a_2(t_i - t_{i-1}) - \beta_2 q_i (t_i - t_{i-1}) + \\ + c_2 p_i (t_i - t_{i-1}) - \frac{d_2}{2}(t_i^2 - t_{i-1}^2) \tag{4.8}$$

Total inventory for the first product Q_1 and for the second product Q_2 are the inventory levels at time $t = 0$ giving the initial inventory levels for the whole inventory cycle, given by

Eq. 4.9 and Eq. 4.10 which can be also found by Eq. 4.13 and Eq. 4.14.

$$\begin{aligned} I^1(0) = Q^1 = &\frac{a_1 - \beta_1 p_1 + c_1 q_1}{\theta_1} e^{\theta_1 t_1} - \frac{d_1 t_1}{\theta_1} e^{\theta_1 t_1} + \frac{d_1}{\theta_1^2} e^{\theta_1 t_1} \\ &- \frac{a_1 - \beta_1 p_1 + c_1 q_1}{\theta_1} - \frac{d_1}{\theta_1^2} + \frac{d_1 t_0}{\theta_1} + I^1(t_1) e^{\theta_1 t_1} \end{aligned} \tag{4.9}$$

$$\begin{aligned} I^2(0) = Q^2 = &\frac{a_2 - \beta_2 q_1 + c_1 p_1}{\theta_2} e^{t_1} - \frac{d_2 t_1}{\theta_2} e^{t_1} + \frac{d_2}{\theta_2^2} e^{\theta_2 t_1} \\ &- \frac{a_2 - \beta_2 q_1 + c_2 p_1}{\theta_2} - \frac{d_2}{\theta_2^2} + \frac{d_2 t_0}{\theta_2} + I^2(t_1) e^{\theta_2 t_1} \end{aligned} \tag{4.10}$$

$$\begin{aligned} Q^1 = \sum_{i=1}^{N} &\frac{a_1 - \beta_1 p_i + c_1 q_i}{\theta_1} e^{\theta_1 (t_i)} - \frac{d_1 t_i e^{\theta_1 (t_i)}}{\theta_1} \\ &+ \frac{d_1 e^{\theta_1 (t_i)}}{\theta_1^2} - \frac{a_1 - \beta_1 p_i + c_1 q_i}{\theta_1} e^{\theta_1 t_{i-1}} - \frac{d_1}{\theta_1^2} e^{\theta_1 t_{i-1}} + \frac{d_1 t_{i-1}}{\theta_1} e^{\theta_1 t_{i-1}} \end{aligned} \tag{4.11}$$

$$\begin{aligned} Q^2 = \sum_{i=1}^{N} &\frac{a_2 - \beta_2 q_i + c_2 p_i}{\theta_2} e^{\theta_2 (t_i)} - \frac{d_2 t_i e^{\theta_2 (t_i)}}{\theta_2} \\ &+ \frac{d_2 e^{\theta_2 (t_i)}}{\theta_2^2} - \frac{a_2 - \beta_2 q_i + c_2 p_i}{\theta_2} e^{\theta_2 t_{i-1}} - \frac{d_2}{\theta_2^2} e^{\theta_2 t_{i-1}} + \frac{d_2 t_{i-1}}{\theta_2} e^{\theta_2 t_{i-1}} \end{aligned} \tag{4.12}$$

Inventory equations can also be written as follows,

$$\begin{aligned} I^1(j) = \sum_{i=j+1}^{N} &\frac{a_1 - \beta_1 p_i + c_1 q_i}{\theta_1} e^{\theta_1 (t_i)} - \frac{d_1 t_i e^{\theta_1 (t_i)}}{\theta_1} \\ &+ \frac{d_1 e^{\theta_1 (t_i)}}{\theta_1^2} - \frac{a_1 - \beta_1 p_i + c_1 q_i}{\theta_1} e^{\theta_1 t_{i-1}} - \frac{d_1}{\theta_1^2} e^{\theta_1 t_{i-1}} + \frac{d_1 t_{i-1}}{\theta_1} e^{\theta_1 t_{i-1}} \end{aligned} \tag{4.13}$$

$$\begin{aligned} I^2(j) = \sum_{i=j+1}^{N} &\frac{a_2 - \beta_2 q_i + c_2 p_i}{\theta_2} e^{\theta_2 (t_i)} - \frac{d_2 t_i e^{\theta_2 (t_i)}}{\theta_2} \\ &+ \frac{d_2 e^{\theta_2 (t_i)}}{\theta_2^2} - \frac{a_2 - \beta_2 q_i + c_2 p_i}{\theta_2} e^{\theta_2 t_{i-1}} - \frac{d_2}{\theta_2^2} e^{\theta_2 t_{i-1}} + \frac{d_2 t_{i-1}}{\theta_2} e^{\theta_2 t_{i-1}} \end{aligned} \tag{4.14}$$

In two product system total revenue comes from both the first product and the second product. As a consequence, we evaluate revenue values separately for the products. Total revenue is the sum of the two different values gained from the products. In each case, revenue is the multiplication of demand with its corresponding price.

Revenue for the first and second product

$$
\begin{aligned}
Revenue^1 &= \sum_{i=1}^{N} \int_{t_{i-1}}^{t_i} (a_1 - \beta_1 p_i + c_1 q_i - d_1 t) p_i dt \\
&= \sum_{i=1}^{N} a_1 p_i (t_i - t_{i-1}) - \beta_1 p_i^2 (t_i - t_{i-1}) + \\
&+ c_1 q_i p_i (t_i - t_{i-1}) - \frac{d_1 p_i}{2} (t_i^2 - t_{i-1}^2) \qquad (4.15) \\
Revenue^2 &= \sum_{i=1}^{N} \int_{t_{i-1}}^{t_i} (a_2 - \beta_2 q_i + c_2 p_i - d_2 t) q_i dt \\
&= \sum_{i=1}^{N} a_2 q_i (t_i - t_{i-1}) - \beta_2 q_i^2 (t_i - t_{i-1}) + \\
&+ c_2 p_i q_i (t_i - t_{i-1}) - \frac{d_2 q_i}{2} (t_i^2 - t_{i-1}^2) \qquad (4.16)
\end{aligned}
$$

Total revenue is sum of revenue gained from first and second product

$$
\begin{aligned}
Total\, Revenue &= \sum_{i=1}^{N} a_1 p_i (t_i - t_{i-1}) - \beta_1 p_i^2 (t_i - t_{i-1}) + \\
&+ c_1 q_i p_i (t_i - t_{i-1}) - \frac{d_1 p_i}{2} (t_i^2 - t_{i-1}^2) + \\
&+ \sum_{i=1}^{N} a_2 q_i (t_i - t_{i-1}) - \beta_2 q_i^2 (t_i - t_{i-1}) + \\
&+ c_2 p_i q_i (t_i - t_{i-1}) - \frac{d_2 q_i}{2} (t_i^2 - t_{i-1}^2) \qquad (4.17)
\end{aligned}
$$

We evaluate cost values with the same procedure used in single product system. Total cost for the whole inventory cycle is addition of cost values from first and second products.

$$C(t_N) = (k_1 + C_1 Q_1)\overline{OF_1} + \underline{h_1}\overline{I_1} + (k_2 + C_2 Q_2)\overline{OF_2} + \underline{h_2}\overline{I_2}$$

We assume cycle times equal to each other for both of the products where

$$\overline{OF_1} = \frac{1}{t_N} \quad and \qquad \overline{OF_2} = \frac{1}{t_N}$$

Total holding cost is sum of the costs resulting from first and second product. By using

Eq. 4.3and Eq. 4.4,

$$\begin{aligned}
\underline{h_1}\overline{I_1} = \frac{h_1}{t_N}\sum_{i=1}^{N}\big[&- \frac{a_1 - \beta_1 p_i + c_1 q_i}{\theta_1^2}(1 - e^{\theta_1(t_i - t_{i-1})}) + \frac{d_1 t_i}{\theta_1^2}e^{\theta_1(t_i - t_{i-1})} \\
&- \frac{d_1}{\theta_1^3}e^{\theta_1(t_i - t_{i-1})} - \frac{a_1 - \beta_1 p_i + c_1 q_i}{\theta_1}(t_i - t_{i-1}) \\
&+ \frac{d_1}{2\theta_1}(t_i^2 - t_{i-1}^2) - \frac{d_1}{\theta_1^2}(t_i - t_{i-1}) + \frac{I^1(t_i)}{-\theta_1}(1 - e^{\theta_1(t_i - t_{i-1})})\big]
\end{aligned} \tag{4.18}$$

$$\begin{aligned}
\underline{h_2}\overline{I_2} = \frac{h_2}{t_N}\sum_{i=1}^{N}\big[&- \frac{a_2 - \beta_2 q_i + c_2 p_i}{\theta_2^2}(1 - e^{\theta_2(t_i - t_{i-1})}) + \frac{d_2 t_i}{\theta_2^2}(1 - e^{\theta_2(t_i - t_{i-1})} \\
&- \frac{d_2}{\theta_2^3}(1 - e^{\theta_2(t_i - t_{i-1})}) - \frac{a_2 - \beta_2 q_i + c_2 p_i}{\theta_2}(t_i - t_{i-1}) \\
&+ \frac{d_2}{2\theta_2}(t_i^2 - t_{i-1}^2) - \frac{d_2}{\theta_2^2}(t_i - t_{i-1}) + \frac{I^2(t_i)}{-\theta_2}(1 - e^{\theta_2(t_i - t_{i-1})})\big]
\end{aligned} \tag{4.19}$$

Profit is the difference between total revenue and total cost. So profit function for the first and second product are given in Eq. 4.20 and Eq. 4.21 where total profit is sum of $\pi_1^{(N)}$ and $\pi_2^{(N)}$, $\pi^{(N)} = \pi_1^{(N)} + \pi_2^{(N)}$.

$$\begin{aligned}
\pi_1^{(N)} = \frac{1}{t_N}\Bigg[\sum_{i=1}^{N}\big(&a_1 p_i(t_i - t_{i-1}) - \beta_1 p_i^2(t_i - t_{i-1}) + \\
&+ c_1 q_i p_i(t_i - t_{i-1}) - \frac{d_1 p_i}{2}(t_i^2 - t_{i-1}^2) \\
&- h_1[-\frac{a_1 - \beta_1 p_i + c_1 q_i}{\theta_1^2}(1 - e^{\theta_1(t_i - t_{i-1})}) + \frac{d_1 t_i}{\theta_1^2}e^{\theta_1(t_i - t_{i-1})} \\
&- \frac{d}{\theta_1^3}(1 - e^{\theta_1(t_i - t_{i-1})}) - \frac{a_1 - \beta_1 p_i + c_1 q_i}{\theta_1}(t_i - t_{i-1}) + \frac{d_1}{2\theta_1}(t_i^2 - t_{i-1}^2) \\
&- \frac{d_1}{\theta_1^2}(t_i - t_{i-1}) + \frac{I^1(t_i)}{-\theta_1}(1 - e^{\theta_1(t_i - t_{i-1})})]\big) - (k_1 + C_1 Q_1)\Bigg]
\end{aligned} \tag{4.20}$$

$$\pi_2^{(N)} = \frac{1}{t_N}\Big[\sum_{i=1}^{N}\big(a_2 q_i(t_i - t_{i-1}) - \beta_2 q_i^2(t_i - t_{i-1}) +$$
$$+ c_2 q_i p_i(t_i - t_{i-1}) - \frac{d_2 q_i}{2}(t_i^2 - t_{i-1}^2)$$
$$- h_2[-\frac{a_2 - \beta_2 q_i + c_2 p_i}{\theta_2^2}(1 - e^{\theta_2(t_i - t_{i-1})}) + \frac{d_2 t_i}{\theta_2^2}(1 - e^{\theta_2(t_i - t_{i-1})})$$
$$- \frac{d_2}{\theta_2^3}(1 - e^{\theta_2(t_i - t_{i-1})}) - \frac{a_2 - \beta_2 q_i + c_2 p_i}{\theta_2}(t_i - t_{i-1}) + \frac{d_2}{2\theta_2}(t_i^2 - t_{i-1}^2)$$
$$- \frac{d_2}{\theta_1^2}(t_i - t_{i-1}) + \frac{I^2(t_i)}{-\theta_2}(1 - e^{\theta_2(t_i - t_{i-1})})]\big) - (k_2 + C_2 Q_2)\Big] \qquad (4.21)$$

Model Formulation of the Problem

In the model, price adjustments in any time interval effect both of the products. Hence model behave as if there is a single product in the system. Model formulation is very close to the form of single product, only difference is profit function. Price adjustment in one product affects the other product's demand value. As a result main profit function includes both profit values coming from products with their associated demands. We use two stage system, firstly second stage is carried out for a given number of price adjustment for both of the products. During the stage, we obtain profit values for the given number of price values, and then we use these values at the first stage of the model with their associated price change cost values, K(N). As a result,we can both maximize profit and also optimize number of price adjustments. In this model number of price adjustments and also inventory cycle time is same for both of the products. If we denote number of price adjustment for the first product as N_1 and for the second product N_2, we take N_1 equal to N_2 and we denote by N in the given model. Hence we use same time array and we evaluate profit values for both of the products for each time interval.

For a given time array that denotes the timings of the price changes,the optimal prices are found as given in Theorem 4.1. However finding the optimal times are not as easy. Thus, we use heuristic algorithms to find the optimal price changing times. As in the single product case, we start from second stage. For a given number of time setting, we evaluate price arrays giving highest profit by the heuristic methods. Then, we use these given number of price settings and their associated profits in the first stage of the model in order to obtain best profit giving number of price and it's corresponding profit,timing and pricing arrays. Heuristic approach is explained in the following section.

$$Stage1: \tag{4.22}$$

$$\pi^* = max \quad \pi^{(N)} - K(N) \tag{4.23}$$

$$Stage2: \tag{4.24}$$

$$\begin{aligned}
\pi^{(N)} = & \frac{1}{t_N}\sum_{i=1}^{N}\big(a_1 p_i(t_i - t_{i-1}) - \beta_1 p_i^2(t_i - t_{i-1}) + \\
& + c_1 q_i p_i(t_i - t_{i-1}) - \frac{d_1 p_i}{2}(t_i^2 - t_{i-1}^2) \\
& - h_1[-\frac{a_1 - \beta_1 p_i + c_1 q_i}{\theta_1^2}(1 - e^{\theta_1(t_i - t_{i-1})}) + \frac{d_1 t_i}{\theta_1^2}(1 - e^{\theta_1(t_i - t_{i-1})}) \\
& - \frac{d}{\theta_1^3}(1 - e^{\theta_1(t_i - t_{i-1})}) - \frac{a_1 - \beta_1 p_i + c_1 q_i}{\theta_1}(t_i - t_{i-1}) + \frac{d_1}{2\theta_1}(t_i^2 - t_{i-1}^2) \\
& - \frac{d_1}{\theta_1^2}(t_i - t_{i-1}) - \sum_{j=i+1}^{N}\{\frac{a_1 - \beta_1 p_j + c_1 q_j}{\theta_1}e^{\theta_1(t_j)} - \frac{d_1 t_j e^{\theta_1(t_j)}}{\theta_1} \\
& + \frac{d_1 e^{\theta_1(t_j)}}{\theta_1^2} - \frac{a_1 - \beta_1 p_j + c_1 q_j}{\theta_1}e^{\theta_1 t_{j-1}} - \frac{d_1}{\theta_1^2}e^{\theta_1 t_{j-1}} - \frac{d_1 t_{j-1}}{\theta_1}e^{\theta_1 t_{i-1}})(1 - e^{\theta_1(t_i - t_{i-1})})\}] \\
& + a_2 q_i(t_i - t_{i-1}) - \beta_2 q_i^2(t_i - t_{i-1}) + c_2 q_i p_i(t_i - t_{i-1}) - \frac{d_2 q_i}{2}(t_i^2 - t_{i-1}^2) \\
& - h_2[-\frac{a_2 - \beta_2 q_i + c_2 p_i}{\theta_2^2}(1 - e^{\theta_2(t_i - t_{i-1})}) + \frac{d_2 t_i}{\theta_2^2}(1 - e^{\theta_2(t_i - t_{i-1})}) \\
& - \frac{d_2}{\theta_2^3}(1 - e^{\theta_2(t_i - t_{i-1})}) - \frac{a_2 - \beta_2 q_i + c_2 p_i}{\theta_2}(t_i - t_{i-1}) + \frac{d_2}{2\theta_2}(t_i^2 - t_{i-1}^2) \\
& - \frac{d_2}{\theta_1^2}(t_i - t_{i-1}) - \sum_{j=i+1}^{N}\{\frac{a_2 - \beta_2 q_j + c_2 p_j}{\theta_2}e^{\theta_2(t_j)} - \frac{d_2 t_j e^{\theta_2(t_j)}}{\theta_2} \\
& + \frac{d_2 e^{\theta_2(t_j)}}{\theta_2^2} - \frac{a_2 - \beta_2 q_j + c_2 p_j}{\theta_2}e^{\theta_2 t_{j-1}} - \frac{d_2}{\theta_2^2}e^{\theta_2 t_{j-1}} \\
& - \frac{d_2 t_{j-1}}{\theta_2}e^{\theta_2 t_{j-1}})(1 - e^{\theta_2(t_i - t_{i-1})})\}]\big) - (k_2 + C_2 Q_2) - (k_1 + C_1 Q_1)
\end{aligned} \tag{4.25}$$

$$s.t \quad D_i^1 \geq 0 \quad and \quad D_i^2 \geq 0 \qquad \forall i \in i = 1, \cdots, N \tag{4.26}$$

$$t_{i-1} - t_i \leq 0 \qquad \forall i \in i = 1, \cdots, N \tag{4.27}$$

$$t_N \geq 0 \tag{4.28}$$

Theorem 4.1 *For a given set of t_i values, the optimal prices p_i^* and q_i^* are found using the result of first order derivative of profit function given in Eq. 4.25. Profit takes its maximum value at the given p_i^* and q_i^*. π is a function which is twice differentiable at p_i and q_i and* *satisfies* $\frac{\partial^2\pi}{\partial p_i^2} < 0, \frac{\partial^2\pi}{\partial q_i^2} < 0$.*Hence* π *has local maximum point at* p_i *and* q_i*, which satisfies both Eq .(4.32) and Eq .(4.33).*

$$\begin{aligned}
\frac{\partial\pi}{\partial p_i} &= a_1(t_i - t_{i-1}) - 2\beta_1 p_i(t_i - t_{i-1}) + c_1 q_i(t_i - t_{i-1}) + c_2 q_i(t_i - t_{i-1}) \\
&- \frac{d_1(t_i^2 - t_{i-1}^2)}{2} - h_1\Big(\frac{\beta_1(1 - e^{\theta_1(t_i - t_{i-1})}}{\theta_1^2} + \frac{\beta_1(t_i - t_{i-1})}{\theta_1}\Big) - \frac{\partial Q^1}{\partial p_i}C_1 \\
&- h_2\Big(\frac{-c_2(1 - e^{\theta_2(t_i - t_{i-1})})}{\theta_2^2} - \frac{c_2(t_i - t_{i-1})}{\theta_2}\Big) - \frac{\partial Q^2}{\partial p_i}C_2 \qquad (4.29) \\
\frac{\partial\pi}{\partial q_i} &= a_2(t_i - t_{i-1}) - 2\beta_2 q_i(t_i - t_{i-1}) + c_2 p_i(t_i - t_{i-1}) + c_1 p_i(t_i - t_{i-1}) \\
&- \frac{d_2(t_i - t_{i-1})}{2} - h_2\Big(\frac{\beta_2(1 - e^{\theta_2(t_i - t_{i-1})}}{\theta_2^2} + \frac{\beta_2(t_i - t_{i-1})}{\theta_2}\Big) - \frac{\partial Q^2}{\partial q_i}C_2 \\
&- h_1\Big(\frac{-c_1(1 - e^{\theta_1(t_i - t_{i-1})})}{\theta_1^2} - \frac{c_1(t_i - t_{i-1})}{\theta_1}\Big) - \frac{\partial Q^1}{\partial q_i}C_1 \qquad (4.30) \\
\frac{\partial Q^1}{\partial p_i} &= \frac{\beta_1(e^{\theta_1(t_{i-1})} - e^{\theta_1(t_i)})}{\theta_1} \\
\frac{\partial Q^2}{\partial p_i} &= \frac{c_2}{\theta_2}(e^{\theta_2(t_i)} - e^{\theta_2(t_{i-1})}) \\
\frac{\partial Q^1}{\partial q_i} &= \frac{\beta_2(e^{\theta_2 t_{i-1}} - e^{\theta_2(t_i)})}{\theta_2} \\
\frac{\partial Q^2}{\partial q_i} &= \frac{c_1}{\theta_1}(e^{\theta_1(t_i)} - e^{\theta_1(t_{i-1})})
\end{aligned}$$

$$(4.31)$$

$$\begin{aligned}
p_i = \frac{a_1}{2\beta_1} + \frac{c_1 q_i}{2\beta_1} - \frac{d_1(t_i + t_{i-1})}{2\beta_1} - h_1\Big[\frac{1 - e^{\theta_1(t_i - t_{i-1})}}{\theta_1^2 2(t_i - t_{i-1})} + \frac{1}{\theta_1}\Big] \\
+ \frac{c_2 q_i}{2\beta_1} - h_2\Big[-\frac{c_2}{\theta_2^2}\frac{1 - e^{\theta_2(t_i - t_{i-1})}}{2\beta_1(t_i - t_{i-1})} \\
- \frac{c_2}{\theta_2 2\beta_1}\Big] - C_2\frac{c_2}{\theta_2}\frac{e^{\theta_2 t_i} - e^{\theta_2 t_{i-1}}}{2\beta_1(t_i - t_{i-1})} - \frac{c_1}{\theta_1}\frac{(e^{\theta_1 t_{i-1}} - e^{\theta_1 t_i})}{t_i - t_{i-1}} \qquad (4.32)
\end{aligned}$$

$$
\begin{aligned}
q_i = {} & \frac{a_2}{2\beta_2} + \frac{c_2 p_i}{2\beta_2} - \frac{d_2(t_i + t_{i-1})}{2\beta_2} - h_2[\frac{1 - e^{\theta_2(t_i - t_{i-1})}}{\theta_2^2 2(t_i - t_{i-1})} + \frac{1}{\theta_2}] \\
& + \frac{c_1 p_i}{2\beta_2} - h_1[-\frac{c_1}{\theta_1^2}\frac{1 - e^{\theta_1(t_i - t_{i-1})}}{2\beta_2(t_i - t_{i-1})} \\
& - \frac{c_1}{\theta_1 2\beta_2}] - C_1\frac{c_1}{\theta_1}\frac{e^{\theta_1 t_i} - e^{\theta_1 t_{i-1}}}{2\beta_2(t_i - t_{i-1})} - \frac{c_2}{\theta_2}\frac{(e^{\theta_2 t_{i-1}} - e^{\theta_2 t_i})}{t_i - t_{i-1}}
\end{aligned} \tag{4.33}
$$

4.1.1 Solution Methodology

Genetic Algorithm

Solution method for the double product system is nearly same with the single product system, when price adjustments are done at the same intervals for each product. We find time arrays by the heuristic algorithms and best profit giving ones selected as the optimal solution. The only difference is the functions used throughout the algorithm such that profit, price and also inventory. Two product heuristic algorithms are created with the same idea in single product, hence only distinct points are highlighted in the following parts.

Representation of the Chromosomes

We use chromosomes with real values, which gives time adjustment points. In our case, n_1 and n_2 are same; as a result only one input parameter is taken as the number of price change which is denoted by N and t_N is the replenishment point for both products. Hence while generating chromosomes, at each iteration end point is taken as t_N. For instance if $t_N = 20$ and $N = 8$, then randomly generated chromosome can be as in Figure 4.2.

Chromosome	0	2,45	15,3	18,2	19,01	7,6	6,78	18,54	18,9	20

Figure 4.2: Illustration of Chromosomes

Initial Solution Generation

We generate initial solutions with the randomization method, where only input is given as the number of price change and the replenishment point. As in the single product system, the important point is population size. If too many chromosomes are generated,this may lead to excessive diversification and unnecessary usage of computational time. We try population sizes of 30,50 and 100 in order to decide on the best size, and choose population size of 30 as in the single product system.

Mutation

We apply one point mutation which is selecting randomly one point along the chromosome and changing its value by a randomized number. In double product system, mutation operator changes only one component of genome, the reason is that mutation of both parts of the genome could be harmful. Because beneficial effect of one mutation can be neutralized by the other mutation. The mutation probability is 0.8 which is very high compared to normal genetic algorithm approaches. The reason is purely experimental; we obtain better performance in these circumstances. Mutation operator is uniform as in single product case, hence there is no need to visualize the mutation.

Crossover

Crossover mechanism is same with the single product case, where we use variable length crossover operator giving us the flexibility to search along the time line with N different points. Alleles are changed between the chromosomes from the crossover cut points.After crossover and mutation, we apply selection mechanism in order to obtain best offspring.

Selection

We try deterministic and roulette wheel selection methods; and decide on that best profit giving values are obtained by the roulette wheel method. After the application of roulette wheel, next generation is formed with randomly selected pairs having higher probability of selection due to having larger ratio of profit in the population.

Simulated Annealing

Solution and Neighbour Representation

Both double and single product systems are based on the same idea of generating initial solution. Arrays with randomized numbers are the initial time solutions and in order to have a neighbour firstly an index is assigned, then the time number is changed into another random one. If the objective function value of the new array is greater than the current one, then it is directly our neighbour. There are some situations when while the objective function value smaller but it is also accepted as our another solution. Acceptance value is used in order to decide whether or not take the array in to neighbour set with the same procedure in single product algorithm.As the temperature value decreases, probability of acceptance also diminishes.

Temperature

As a cooling schedule, exponential decay is used. Double product systems are much complicated by their nature. Because each part of the solution can effect the other, the problem becomes a wicked problem which is by definition, changes its objective while assessing the problem itself. The chaotic behaviour of the solution function requires more elaborate search, which can be done at increased temperatures. Keeping that in mind and experimental results, temperature is set to 10.

Chapter 5

COMPUTATIONAL STUDIES

In this chapter, we give details about the parameter settings, heuristic algorithm and also sensitivity analysis results.In Section 5.1, parameters for both single and double product system can be found.In Section 5.1, we explain the parameter settings for the genetic algorithm and also simulated algorithm. In Section 5.2.3, we give detailed explanation for the computational results.

5.1 Data Generation

For both single and double product system one set of parameters is chosen as a basis for the experimental studies. Demand function parameters, inventory costs, fixed costs and also price change costs are included in this parameter set. In Table 5.1 and Table 5.2 we give the parameters and their values with their related abbreviations. Demand both depends on time and price; as price increases demand decreases. Due to perishability, as time passes product loses its freshness; so demand decreases. These factors are all included in demand by "β" and "d" values. In double product system we include "c_1" and "c_2" giving the substitution property of two products; if we increase one the product's price then, demand for the second product increases.This correlation is maintained in demand by using "c_1" and "c_2". Order costs are given for each replenishment cycle, closely related with the total inventory for the whole cycle. As a result, if we give an order by size "Q" at the end of "t_N", a total cost of "CQ" incurs. In double product system, unit order costs are given by "C_1" and "C_2". Fixed costs are independent of size of the inventory, hence whenever an order is given a total fixed cost takes place with an amount of "k" in single system or "k_1" and "k_2" in double product system. Decay coefficients determines the perishability ratio given by "θ" in single product system and in double product system given by "θ_1" and "θ_2".

Table 5.1: Single Product System Parameters

Parameter	Abbreviation	Unit value
Time Dependency	d	0.1
Market Potential	a	100
Price Sensitivity	β	0.3
Unit Holding Cost	h	1
Decay Coefficient	θ	0.01
Fixed Cost	k	500
Unit Order Cost	C	10
Price Change Cost	f	10

Table 5.2: Double Product System Parameters

Parameter	Abbreviation	Unit Value
1^{st} Product Market Potential	a_1	100
2^{nd} Product Market Potential	a_2	100
1^{st} Product Sensitivity	β_1	0.3
2^{nd}Product Price Sensitivity	β_2	0.3
1^{st} Product Unit Order Cost	C_1	10
2^{nd} Product Unit Order Cost	C_2	10
1^{st} Product Price Dependency	c_1	0.1
2^{nd} Product Price Dependency	c_2	0.1
1^{st} Product Time Dependency	d_1	0.1
2^{nd} Product Time Dependency	d_2	0.1
1^{st} Product Price Change Cost	f_1	10
2^{nd} Product Price Change Cost	f_2	10
1^{st} Product Fixed Cost	k_1	500
2^{nd} Product Fixed Cost	k_2	500
1^{st} Product Unit Holding Cost	h_1	1
2^{nd} Product Unit Holding Cost	h_2	1
1^{st} Product Decay Coefficient	θ_1	0.01
2^{nd} Product Decay Coefficient	θ_2	0.01

5.2 Parameter Settings for the Algorithms

For both genetic algorithm and simulated annealing in order to achieve best objective function value, different algorithm parameter values are tested; such as temperature, selection, population size or cooling schedule. It is important to decide on the best parameters which directly affects the performance of the heuristic algorithms, so well tuning is a must. Below you can find parameter setting results for both single and double product system.

5.2.1 Parameter Setting for Genetic Algorithm

Population size

In genetic algorithm population size, selection methods, crossover and mutation probabilities are the main parameters that we deal with. We try different population sizes in order to achieve best results by taking into account some specific informations. Such that, if larger population sizes are used, this can result in high amount of diversification or small population may result in inadequate search of solution space which are undesired conditions for the heuristic algorithms. Hence, diversification and intensification should be well managed. We try population sizes of 30, 50 and 100 for both single and double product systems; results are given in Table 5.3

Table 5.3: Population Size Determination for Single and Double Product System

Population Size	Profit Single	Profit Double
Popsize 30	7755,00	11377,00
Popsize 50	7754,78	11377,00
Popsize 100	7755,00	11377,00

After the application of different population sizes, we've decided to use population size of 30, which is enough for the genetic algorithm for both single and double product system. We do not choose 50 or 100, because best results can be achieved at lower sizes which leads us having computational efficiency and having less CPU time by maintaining balance between diversification and intensification.

Selection

After crossover and mutation applied, we need to form off-springs in order to maintain gene continuity giving higher profit results. We try two methods for both single and double product system; roulette wheel and deterministic selection. Compared values for single and double product system are given in Table 5.4; both of which reach the largest amount of profit when we use roulette wheel selection.

Table 5.4: Selection Methods for Single and Double Product System

Selection	Single Profit	Double Profit
Roulette Wheel	7509,40	10953,00
Deterministic	7507,45	10951,00

Mutation

Genetic algorithm diversification is maintained by mutation operator, which can have different values specific to the model. We try different mutation probabilities given in Table 5.5. In single system mutation probability is set to 0.07 and in double product system it is set to 0.8, which give better results with respect to other mutation probabilities. We have also tried lower values such as 0.07 for double product system; however we obtained worse results compared to higher probabilities.

Table 5.5: Single and Double Product Mutation Probabilities Comparison

Probability	Profit Single	Probability	Profit Double
0.07	7509,40	0.8	10953,20
0.02	7509,20	0.7	10952,80
0.05	7509,30	0.9	10952,70
0.08	7509,10	0.95	10953,00

5.2.2 Parameter Setting for Simulated Annealing

Temperature

In simulated annealing temperature values provide the necessary intensification and diversification for the problem. If too high temperatures are selected, then simulated annealing will have no difference from the scatter search. Hence fine tuning is a necessary action for us to obtain good results. In Table 5.6 you can find temperature values where at temperature 2 single product system and at 10 double product system take their highest objective values.

Table 5.6: Single Product Simulated Annealing Results at Different Temperatures

Temperature	Profit Single	Profit Double
2	7705,30	11350,00
10	7604,20	11361,00
30	7604,20	11358,00
100	7604,20	11350,78

5.2.3 Analysis of the Single Product System

In this section we share the experimental results obtained by the heuristics for both single and double product system. In addition to the heuristic results, constant pricing values and its comparisons with single and double product system are also given.

Analysis of the Single Product System

Experiments are done on a workstation with an Intel(R) Core(TM)2 Duo processor, 2.53 GHz speed, and 2GB of RAM. Heuristic methods are coded in Matlab 7.12.0. The first case is constant pricing; we only set prices at the beginning of the cycle and do not change again until the proceeding replenishment time. The constant pricing results for the given parameters in Table 5.1 are shown in Table 5.7 and detailed sensitivity results are given in Table 5.14.

Table 5.7: Single Product Constant Pricing Results for the Base Parameters

Parameter	Profit	Decay	Inventory	Price	Holding Cost	Cycle	Revenue
Base Case	7437,80	0,015	146,71	172,25	221,85	3,00	8315,40

Results for the Number of Price Adjustment Change

The number of price change is the main issue in order to see the effects on the profit and also other parameters. Hence, we try different numbers of price change by using the parameters given in Table 5.1 and share the results in Table 5.8 giving profit, order and holding cost and also average price values. Holding cost is the value of keeping inventory throughout the cycle found by $h * \overline{I}$; order cost is the procurement cost of the material which depends on unit procurement cost and initial inventory evaluated by "CQ". Figure 5.1 gives the profit

Table 5.8: Comparisons of optimal pricing strategies regarding N

N	Profit	K(N)	Holding Cost	Order Cost	Total
1	7502,268	10,000	157,853	698,944	866,797
2	7518,194	20,000	131,343	699,350	850,693
3	7521,900	30,000	158,050	658,421	846,471
4	7521,900	40,000	147,550	658,577	846,127
5	7518,595	50,000	140,429	658,681	849,110
6	7518,094	60,000	135,668	658,753	854,421
7	7506,475	70,000	132,102	658,805	860,907
8	7499,263	80,000	129,181	658,587	867,768
9	7491,550	90,000	126,717	658,888	875,605
10	7482,736	100,000	125,501	658,899	884,399

change with respect to number of price adjustments. Profit value increases up to a certain $N = 4$, then starts to decrease; increase in operational and price change costs result in a decrease of profit value. Optimal price change with respect to time at $N = 4$ is given in Figure 5.3 which has a decreasing pattern.In this figure you can also find the constant price result given with red line having cycle time $t_N = 3$. There are two main reasons for the different t_N value; one which is higher initial inventory in multi price case, the other one is lower holding cost. Hence optimal solution tends to keep inventory for a longer time in multi pricing case which leads to longer cycle length. As a result by using lower price values and larger cycle lengths we can obtain higher profits.

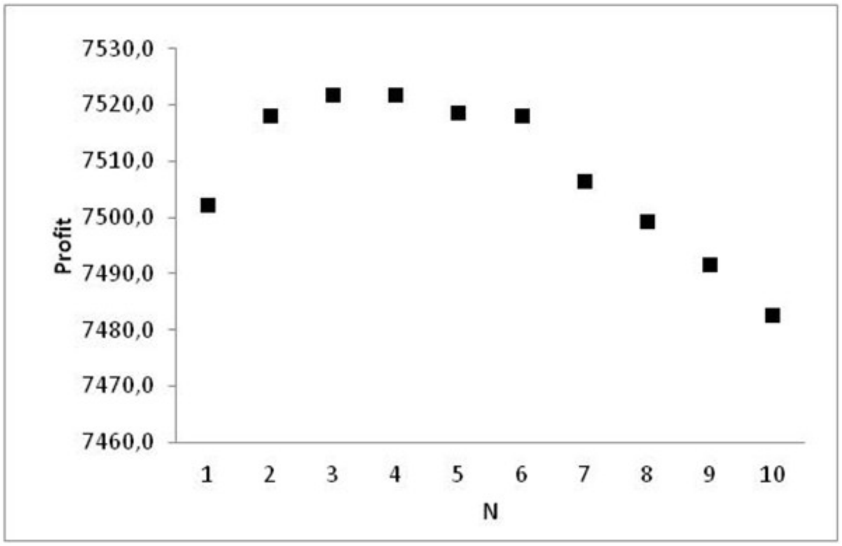


Figure 5.1: Profit Change with respect to N

For each number of price change, average prices are enumerated given in Figure 5.2 which is a comparison between "N" and "$averageprice$". As the number of price change increases, we obtain lower price average. Order and holding costs are given in Figure 5.4 and Figure 5.5; both show a decreasing pattern and take their minimum values at $N = 3$ and $N = 2$. However without considering price change cost, making a conclusion would be wrong. Hence, total change costs are evaluated. Total cost change including price change with respect to N is given in Figure 5.6, showing that up to optimal N^* total cost decreases; profit shows a parallel pattern with the total cost. In our model, the optimal $N^* = 4$ is the point where we obtain maximum profit, 7509.60 where model gets minimum total cost. When we analyze order and holding cost figures and do not consider price change costs, we may come up with a result that as "N" increases holding and order cost decrease. For this reason profit values should increase. However when include price change costs, we would face a total cost change pattern as given by Figure 5.6 having parabolic structure where it gets its minimum value at $N = 4$. As a result in real case profit increases up to some point then starts to decrease due to total cost rise. We can see the effects of $K(N)$ in the following

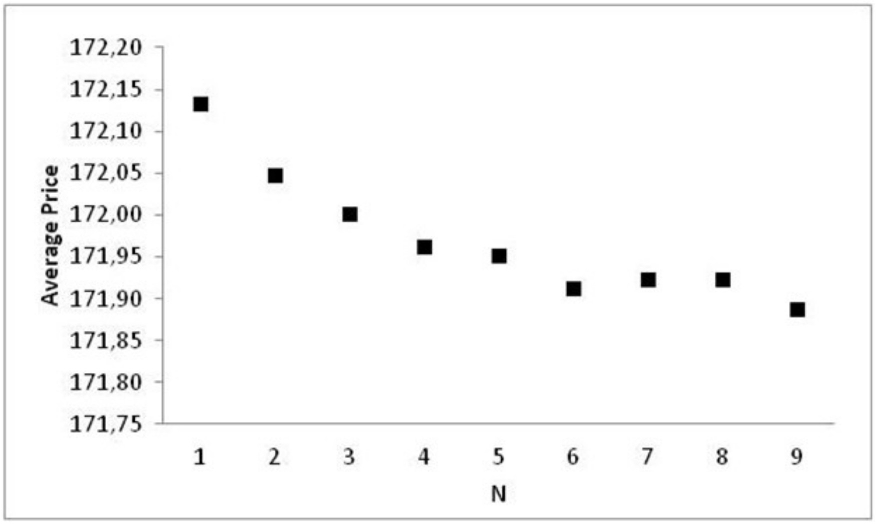


Figure 5.2: Average Price Change with respect to N

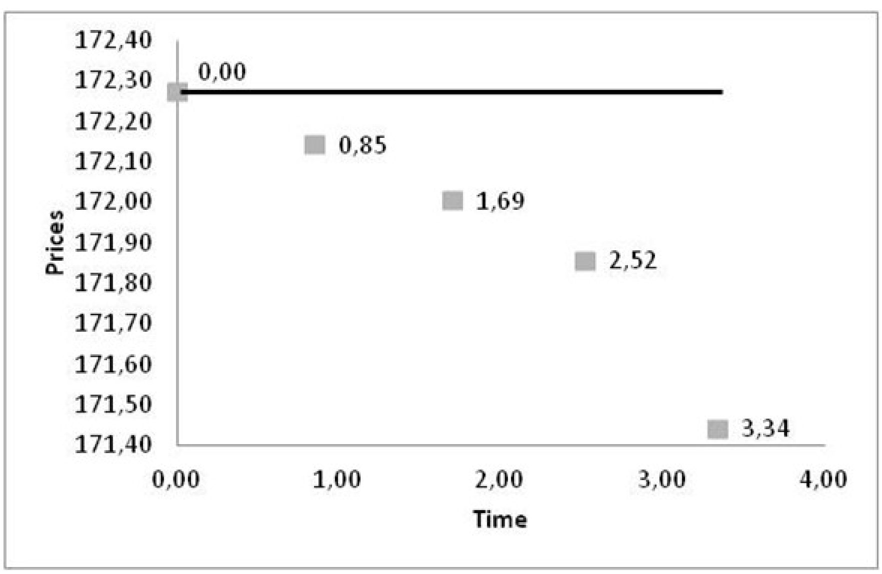


Figure 5.3: Optimal Price Change with respect to Time, N=4

section.

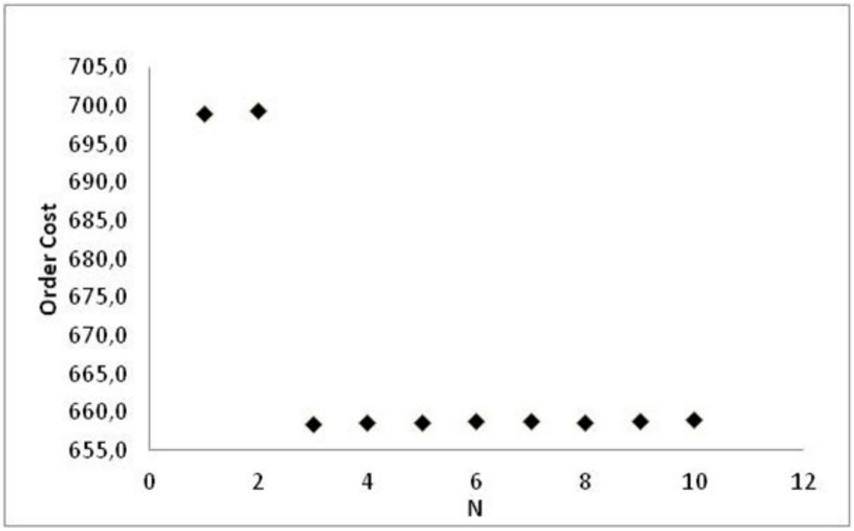


Figure 5.4: Order Cost Change with respect to N

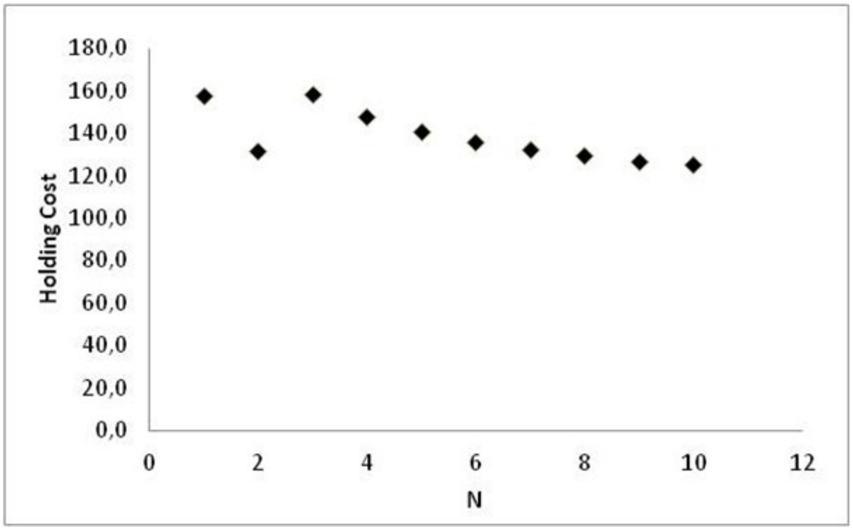


Figure 5.5: Holding Cost Change with respect to N

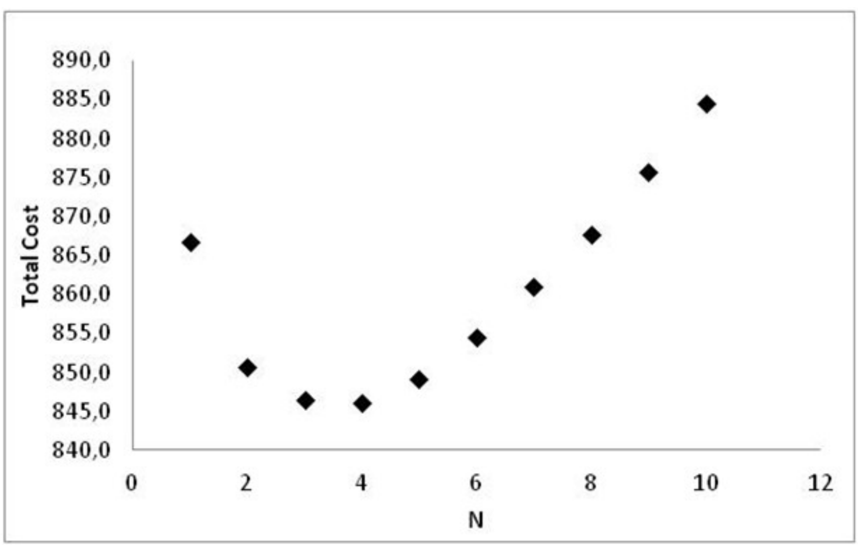


Figure 5.6: Total Cost Change Change with respect to N

Price Change Cost Results

Price change cost is an important parameter that directly effects "N", profit, initial inventory and the other parameters. We make experiments with three different "f", and analyze their results. In these tables "$Profit$" gives the profit values for different "N" and "f". Cycle is the replenishment time for the inventory. At each time interval different optimal price values are found, in tables these prices are given as average. Such that for $N = 4$, "$Price$" is the average of four optimal prices. Inventory gives the initial inventory values used throughout the cycle. Hold gives the inventory holding cost, $h\bar{I}$. Order cost is the procurement cost of inventory which is evaluated by "CQ" where "Q" is the initial inventory.

Figure 5.7 illustrates the profit change at different "f" values. $f = 0$ is the case of continuous pricing, where profit value always increases due to lack of price change cost. On the other hand at $f = 5$, graph shows a parabolic structure with both increments and decrements having maximum value at N^*. At constant pricing case, profit value displays stable pattern. During $f = 50$, the rate of profit decrease is greater than $f = 5$, which is a result of higher price change and total costs (order and holding costs), given in Figure 5.7. Price change at $f = 0$ with respect to optimal times are also given in Figure 5.8 having $N = 30$ and $t_N = 4$, where prices decrease in both simulated and genetic algorithm, even below the optimal constant price by getting higher profits due to lower operational costs like order or holding cost. Here the important point is that at each time price change is applied genetic algorithm or simulated annealing gets lower price than the preceding one which makes us to have higher profit in the end which is a typical result for continuous pricing case. All the results are evaluated with the parameter set given in Table 3.1.

Total cost change with respect to "N" is given in Figure 5.9; at $f = 0$ total cost decreases as "N" increases. Continuous profit increase is the main result of having decreasing total cost. In both $f = 5$ and $f = 50$, graphs have an increasing structure, especially for $f = 50$ there is a sharp increase resulting in rapid profit decrease as shown in Figure 5.7. Average price change has same decreasing structure at all given "f" values,shown in Figure 5.10; all of which are below the optimal constant price. Especially for $f = 5$ and $f = 0$ we obtain higher profits with lower average price given in Table 5.8. Detailed numerical results are shown in Table 5.9, Table 5.10 and Table 5.11 for $f = 0, f = 5$ and $f = 50$.

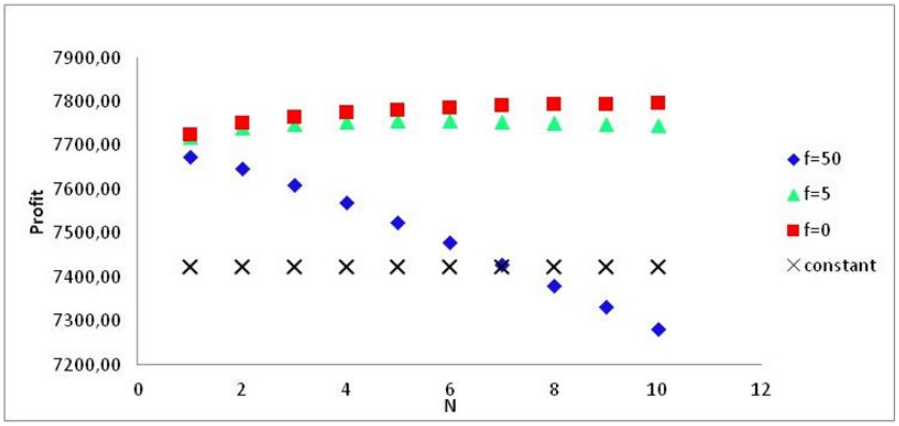


Figure 5.7: Total Profit Change with respect to$f$

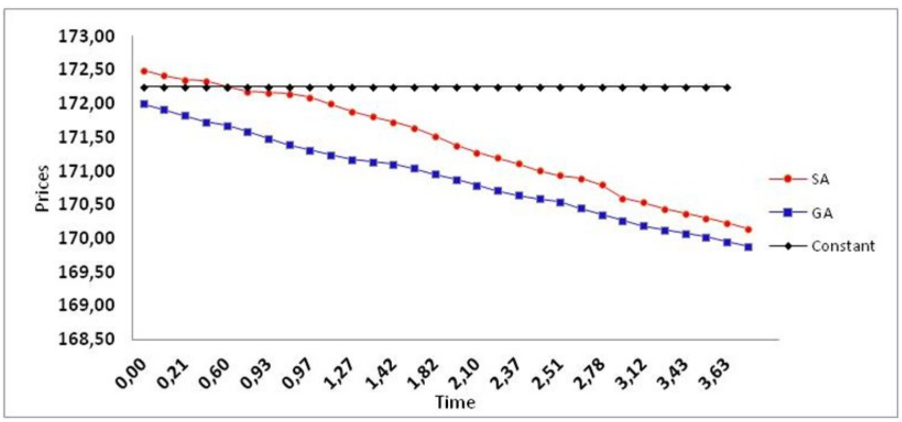


Figure 5.8: Price Change Pattern with Simulated Annealing, Genetic Algorithm and Constant Pricing at $f = 0$, $h = 1$; $t_N = 4$, $N = 30$

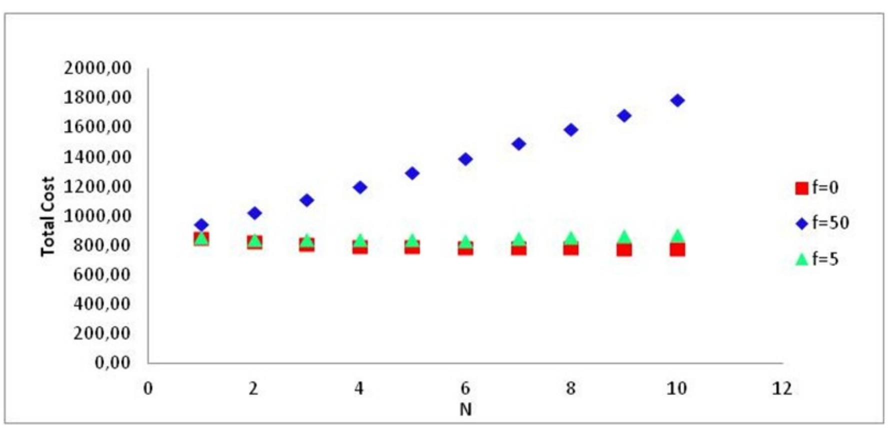


Figure 5.9: Total Cost Change with respect to f

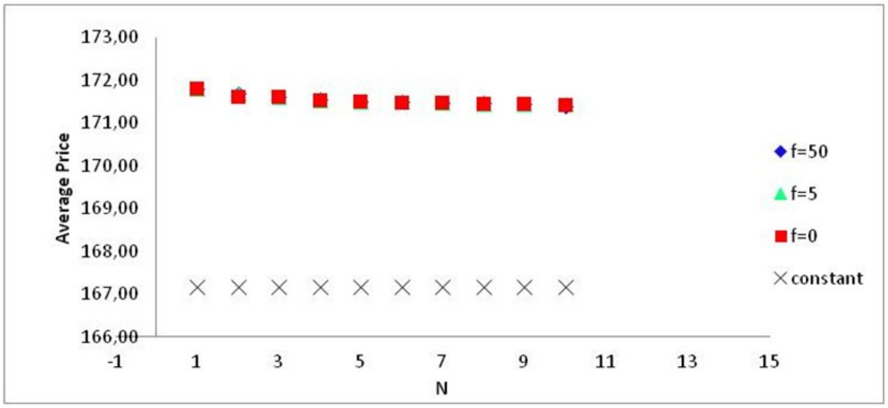


Figure 5.10: Average Price Change with respect to f

Table 5.9: Numerical Results with f=0

N	Profit	Cycle	Average Price	Initial Inventory	Holding Cost	Order Cost	Total Cost
1	7725,82	3,00	171,79	156,15	156,40	692,50	848,90
2	7751,68	3,00	171,60	156,35	130,14	692,89	823,03
3	7765,79	4,00	171,59	209,44	156,58	652,34	808,92
4	7775,99	4,00	171,53	209,50	146,25	652,49	798,74
5	7782,99	4,00	171,50	209,54	139,20	652,60	791,80
6	7787,83	4,00	171,47	209,57	134,25	652,67	786,93
7	7791,44	4,00	171,45	209,59	130,55	652,73	783,28
8	7794,83	4,00	171,44	209,61	130,60	652,76	783,37
9	7796,28	4,00	171,42	209,62	125,62	652,80	778,42
10	7797,92	4,00	171,41	209,63	124,01	652,82	776,83

Table 5.10: Numerical Results with f=5

N	Profit	Cycle	Average Price	Initial Inventory	Holding Cost	K(N)	Order Cost	Total Cost
1	7720,67	3,00	171,80	156,25	156,40	5,00	697,65	859,05
2	7741,38	3,00	171,67	156,37	130,13	10,00	703,19	843,33
3	7750,34	4,00	171,59	209,44	156,60	15,00	667,79	839,39
4	7755,59	4,00	171,53	209,50	146,10	20,00	673,10	839,19
5	7757,24	4,00	171,50	209,54	139,11	25,00	678,35	842,46
6	7756,72	4,00	171,49	209,57	134,47	20,00	683,57	838,03
7	7755,49	4,00	171,46	209,59	130,48	35,00	688,78	854,26
8	7752,71	4,00	171,44	209,60	128,11	40,00	693,96	862,07
9	7749,62	4,00	171,43	209,62	126,00	45,00	699,14	870,14
10	7746,42	4,00	171,42	209,63	123,97	50,00	704,32	878,29

Table 5.11: Numerical Results with f=50

N	Profit	Cycle	Average Price	Initial Inventory	Holding Cost	K(N)	Order Cost	Total Cost
1	7674,32	3,00	171,80	156,25	156,40	50,00	744,00	950,40
2	7648,68	3,00	171,67	156,37	130,13	100,00	795,89	1026,03
3	7611,29	4,00	171,59	209,44	156,59	150,00	806,84	1113,43
4	7570,19	4,00	171,53	209,50	146,05	200,00	858,50	1204,55
5	7525,39	4,00	171,49	209,54	139,29	250,00	910,10	1299,39
6	7478,83	4,00	171,48	209,57	134,19	300,00	961,67	1395,87
7	7430,32	4,00	171,46	209,58	131,26	350,00	1013,21	1494,47
8	7381,91	4,00	171,46	209,60	128,04	400,00	1064,71	1592,75
9	7332,67	4,00	171,44	209,62	125,73	450,00	1116,31	1692,05
10	7282,72	4,00	171,39	209,63	124,23	500,00	1167,81	1792,05

Sensitivity Results

In order to see the effects of parameter change, we apply sensitivity analysis on both genetic and simulated annealing algorithms. Basis parameters are given in Table 3.1. Below you can find Table 5.16,Table 5.15 and Table 5.18 giving sensitivity analysis results. In Table 5.16 and Table 5.19 give the comparison between constant pricing and heuristic algorithms, which is found by $(heuristic\,value - constant\,value)/constant\,price\,value$. In Table 5.17, we share genetic algorithm's sensitivity results compared with the base case, hence how much change is obtained with parameter revise can be discussed. Table 5.17 is formed according to the equation $(new\,parameter\,value - base\,case\,value)/base\,case\,value$. In all tables "Profit" gives the maximum profit reached by heuristic algorithms and constant pricing. "Decay ratio" is the ratio of perished inventory to the total inventory. "Inventory" gives the initial amount of material at the beginning of replenishment cycle. In multi pricing case "Price" is the average of the prices found by the heuristic algorithms for different time intervals. "Holding Cost" is evaluated by $h * \overline{I}$, which is multiplication of average inventory and holding cost giving total holding cost for the whole inventory cycle. "N" is the number of price adjustments. "Cycle length" is the end of the inventory cycle where new order has to be given due to zero on hand inventory. "Revenue" is the total gain found by the multiplication of demand and prices for each time interval.

The constant "a" includes the effects of all factors other than price that affect demand.

Such that if income were to change, the effect of the demand would be represented by a change in the value of "a" and can be reflected graphically as a shift of the demand curve. As we increase "a" profit,revenue and also price per unit increases which is an intuitive result that we expect. However cycle length decreases due to large amount of demand which fosters high amount of sale. As a result reorder time and decay amount decreases by 25%, whereas stock value increases by 50% due to large amount of material need which is given in Table 5.17.

The parameter "β", which is the price sensitivity of material; if β increases, material becomes more price dependent hence the unit profit gain decreases. This issue is directly correlated with price elasticity. If a material has a higher price elasticity constant, than any unit increase in the prices will cause demand decrease in the market. As "β" increases ($\beta = 0.6$), profit gain and total revenue decrease by 50% on average; on the other hand cycle time,decay ratio and inventory keep same. You can find details in Table 5.15.

Fixed set up cost is independent of order size which has to be paid every time we give an order. As a result if its value increases, it directly effects cycle length, inventory and decay amount. As the set up cost increases, cycle time increases by 25% which leads to 25% more inventory in the system. When fixed cost is higher $k = 1000$, the optimal system tries to extend the cycle time in order to prevent high set up expense. When inventory is high but demand is not high as much as the inventory, then decay amount increases by 25%.

The "C" is the unit order cost changing with the total inventory in the system. If the total inventory ordered is high, then the total order cost will also be higher. As the unit value increases $C = 20$, optimal system tries to hold inventory as low as possible. Hence the cycle length and decay ratio decreases by 25%. Revenues with respect to "C" are close to each other but due to the higher variable cost, $C = 20$ has the minimum profit value which is a 7% loss from the basis value. Cycle length decreases by 25% and decay ratio by 25% as "C" increases from 5 to 20. When we analyze the sensitivity result with respect to "d", we figure out that giving values from $0.05 - 0.1$ have very little impact on the profit. The major point here is if we change the time dependency of material "d" between 0.01-0.2, this will result in a lower holding cost in the system which leads to decrease in decay ratio by 1% as given in Table 5.17 and Table 5.15.

Holding cost is the another parameter which have the most influence on the optimal result of the model. When we take $h = 2$, the profit directly gets smaller by 1% due to the total cost increase by 50% on average. In addition to that cycle time also decreases by 25% in order not to carry too much inventory in the system resulting in lower efficiency ; also system tends to adjust prices more than once during the time array so as to get higher profit. When we change $h = 1$ to $h = 2$ cycle time decreases by 25% and inventory decreases by 25%.

Decay constant is the parameter which determines primarily initial inventory and decay amount in the economical model. When we change its value from 0.01 to 0.02 the inventory decreases by 24% ,cycle length decreases by 25%. If we change "θ" to 1, the cycle length decreases by 75% and profit decreases by 8%.

In Genetic Algorithm for all parameter changes, profit values show an improvement compared to the constant pricing case. Most improvement is achieved with the profit when $\beta = 0.6$ and $a = 50$ by 2.5% and 3.2% respectively given in Table 5.16. Price values decrease by 0.1 on average. Inventory held over the cycle increases by 30.1% due to the reorder time extension by 24.8%. Consequently the inventory amount increase by 30.1%%, make us to think that holding cost will also increase. Whereas, we achieve directly an opposite result,a 32% reduction on average, due to the different strategy in the computation of inventory costs; such that finding costs for each time interval explicitly rather than only multiplying initial inventory with holding cost. Using pricing strategy at different time intervals make us to have smaller order cost and larger profit for the whole inventory system. You can see from Table 5.17 that we get noticeable amount of holding cost reduction (from 19.7% to 59.5%) in the system which is the biggest advantage of multi pricing cases.

We can also achieve this result by examining the cycle time values in multi price system; we can achieve on average 24.8% cycle length increase. Decay ratios also rise due to cycle time and inventory increase in the system. Decay ratio increases by 26.2% on average with respect to the constant pricing case. This is an obvious result which can be deduced by just analyzing the inventory rise which is 30.1% and also cycle length increase by 24.8% which contribute to both larger inventory and waiting time in the system; hence larger amount of perishability. Detailed results are shared in Table 5.16.

We also try time values which are equidistant and give their results in Table 5.20. In order to make its comparison between heuristic algorithms and constant pricing, we form

Table 5.13. Equidistant time arrays are used with genetic algorithm and we find out that it gives better results than any other case like simulated, genetic or constant pricing. It gives 1.1361% higher profit value than the constant pricing case for the base parameters. However genetic algorithm with random numbers and simulated annealing gives 1.1307% and 0.48% better result compared to constant pricing values. The algorithm having equally spaced time interval performs its maximum profit change at $a = 50$ and $\beta = 0.6$ by 3.15% and 2.5% respectively. In addition it gives its lowest profit at $k = 0$. Genetic algorithm takes its maximum profit value change at $a = 50$ and its lowest change at $a = 200$; for simulated annealing maximum change is obtained at $f = 5$ by 1.41% and lowest change at $d = 0.8$.

Genetic algorithm results given in Table 5.17 point out that we have an improvement in profit values changing from 0.5% to 3.2% and also holding cost reduction by 32%, cycle length increase by 24.8%, average price reduction by 0.1%, rise in decay amount by 26.2%. When we analyze the simulated annealing results in Table 5.19, the improvement ratios are smaller. Profit values take their maximum improvement with respect to the constant pricing sensitivity results at $k = 1000$ by 1.15% and $f = 5$ by 1.59%. For the base case, it provides a profit increase by 0.48%, which is much smaller ratio compared to 1.1% obtained by the genetic algorithm. Genetic algorithm give a rise to profit up to 3.2% at $a = 50$, on the other hand simulated annealing can give a rise of 1.41% at maximum when $f = 5$.

We also examine the profit values by using the equally spaced time lines, a simple heuristic that searches over equally spaced time lines in order to obtain best profit giving time array, and also genetic algorithm having equally spaced time line population. When we compare the values equal interval time line simple heuristic solutions are found to be better than the constant pricing, but worse than the heuristic algorithms. In order to have an insight about the genetic algorithm improvements when we use random population and also equally spaced time line population, we form the Table 5.12. In Table 5.12 you can clearly see that we obtain better objective function value when we use equally spaced time line rather than using only random time line population; when we use the base case parameters, we reach an improvement of 0.69%. In Table 5.13 we give all the sensitivity results with respect to the genetic algorithm, simulated annealing, genetic algorithm with equally spaced time line population, constant pricing and equally spaced time line simple heuristic solutions and they are compared with the constant pricing where $\%difference$ gives the difference from the constant pricing case. Genetic algorithm with equally spaced time line population gives the maximum improvement from constant pricing case by 1.1361%. On the other hand when we use only equally spaced time line simple heuristic, we can reach an improvement of 0.44%. By summing all of these, we can conclude that genetic algorithm with equally spaced time line population outperforms simulated annealing and other solution methods with respect to their objective function values.

Table 5.12: The Comparison of Profit Values between Equal Interval Time Line Simple Heuristic Solution and GA with Equal Interval Time Line Population

Parameter	Equal Interval	Equal Interval with GA	% difference
Base Case	7470,71	7522,30	0,69
a=200	31830,16	31935,77	0,33
a=50	1538,36	1602,65	4,18
k=1000	7355,32	7406,30	0,69
k=0	7723,04	7776,24	0,69
k=250	7532,29	7604,74	0,96
β=0,6	3277,57	3353,25	2,31
β=0,15	15773,17	15860,20	0,55
C=20	6964,39	7014,03	0,71
C=5	7703,37	7776,64	0,95
d=0,2	6916,62	7515,39	8,66
d=0,05	7446,18	7525,71	1,07
d=0	7477,60	7529,11	0,69
f=20	7495,95	7498,76	0,04
f=5	7540,01	7543,14	0,04
h=2	7332,61	7412,11	1,08
h=0.5	7521,07	7601,04	1,06
θ=0.02	7417,39	7511,78	1,27
θ=0.005	7457,38	7528,81	0,96
θ=1	6342,61	6889,41	8,62

Table 5.13: The Comparison of Profit Values for Single Product System

Parameter	Equal Interval	% difference	Equal Interval with GA	% difference	GA	% difference	SA	% difference	Constant
Base Case	7470,71	0,44	7522,30	1,1361	7521,90	1,1307	7473,69	0,48	7437,80
a=200	31830,16	0,06	31935,77	0,3948	31934,07	0,3895	31842,87	0,10	31810,17
a=50	1538,36	-0,98	1602,65	3,1574	1602,56	3,1520	1538,97	-0,94	1553,59
k=1000	7355,32	1,11	7406,30	1,8116	7405,91	1,8062	7358,26	1,15	7274,52
k=0	7723,04	-0,50	7776,24	0,1891	7775,82	0,1837	7726,12	-0,46	7761,56
k=250	7532,29	0,34	7604,74	0,6143	7604,34	0,6090	7535,29	-0,30	7558,31
β=0,6	3277,57	0,19	3353,25	2,5060	3353,07	2,5005	3278,87	0,23	3271,27
β=0,15	15773,17	-0,01	15860,20	0,5378	15859,36	0,5324	15779,47	0,03	15775,36
C=20	6964,39	0,46	7014,03	1,1771	7013,66	1,1718	6967,17	0,50	6932,42
C=5	7703,37	0,17	7776,64	1,1208	7776,22	1,1155	7706,44	0,21	7690,44
d=0,2	6916,62	-6,92	7515,39	1,1399	7514,99	1,1345	6919,38	-6,88	7430,69
d=0,05	7446,18	0,07	7525,71	1,1342	7525,31	1,1288	7449,16	0,11	7441,31
d=0	7477,60	0,44	7529,11	1,1323	7528,71	1,1269	7480,59	0,48	7444,81
f=20	7495,95	0,78	7498,76	0,8196	7498,36	0,8142	7498,94	0,82	7437,80
f=5	7540,01	1,37	7543,14	1,4162	7542,73	1,4108	7543,03	1,41	7437,80
h=2	7332,61	0,68	7412,11	1,7667	7411,72	1,7613	7335,53	0,72	7283,43
h=0.5	7521,07	-0,40	7601,04	0,6547	7600,63	0,6493	7524,07	-0,36	7551,60
θ=0.02	7417,39	-0,11	7511,78	1,1595	7511,38	1,1542	7420,35	-0,07	7425,68
θ=0.005	7457,38	0,18	7528,81	1,1365	7528,41	1,1311	7460,36	0,22	7444,21
θ=1	6342,61	-6,68	6889,41	1,3598	6889,05	1,3544	6345,15	-6,65	6796,99

Table 5.14: Sensitivity Analysis with Constant Pricing

Parameter	Profit	Decay Ratio	Inventory	Price	Holding Cost	Cycle Length	Revenue
Base Case	7437,800	0,015	146,713	172,250	221,850	3,000	8315,400
a=200	31810,171	0,010	194,147	343,801	294,750	2,000	33278,980
a=50	1553,595	0,020	96,659	86,564	147,463	4,000	2076,628
k=1000	7274,517	0,020	194,910	172,428	297,485	4,000	8315,300
k=0	7761,562	0,005	48,294	171,900	73,115	1,000	8315,500
k=250	7558,309	0,010	96,878	172,071	147,067	2,000	8315,400
β=0,6	3271,274	0,015	145,083	86,515	220,835	3,000	4157,450
β=0,15	15775,361	0,015	146,083	343,720	222,357	3,000	16631,000
C=20	6932,423	0,015	145,750	172,250	221,850	3,000	8315,400
C=5	7690,438	0,015	145,750	172,250	221,850	3,000	8315,400
d=0,2	7430,688	0,015	145,527	171,992	221,439	3,000	83071,095
d=0,05	7441,306	0,015	145,861	172,379	222,055	3,000	8319,495
d=0	7444,812	0,015	145,972	172,508	222,261	3,000	8323,691
f=20	7437,800	0,015	145,750	172,250	221,850	3,000	8315,400
f=5	7437,800	0,015	145,750	172,250	221,850	3,000	8315,400
h=2	7283,432	0,010	96,584	172,593	293,231	2,000	8315,200
h=0.5	7551,597	0,020	195,508	171,906	149,200	4,000	8315,500
θ=0.02	7425,679	0,020	97,844	172,071	149,384	2,000	8315,400
θ=0.005	7444,211	0,008	14,656	172,246	219,266	3,000	8315,400
θ=1	6796,989	0,418	82,502	172,011	177,404	1,000	8315,500

Table 5.15: Sensitivity Analysis Results with Genetic Algorithm

Parameter	Profit	Decay Ratio	Inventory	Price	Holding Cost	Cycle Length	Revenue
Base Case	7521,90	0,02	197,08	172,00	147,55	4,00	8317,20
a=200	31934,07	0,01	294,72	343,83	223,37	3,00	33285,78
a=50	1602,56	0,02	122,95	86,22	109,52	5,00	2077,25
k=1000	7405,91	0,02	247,30	172,08	198,27	5,00	8317,20
k=0	7775,82	0,01	48,63	171,95	52,15	1,00	8317,20
k=250	7604,34	0,01	97,69	171,99	87,46	2,00	8317,20
β=0,6	3353,07	0,02	196,71	86,16	158,25	4,00	4158,60
β=0,15	15859,36	0,02	197,17	343,83	158,82	3,11	16634,40
C=20	7013,66	0,01	147,15	171,99	118,20	3,00	8317,20
C=5	7776,22	0,02	226,09	172,03	158,36	4,00	8317,20
d=0,2	7514,99	0,02	196,62	171,97	157,63	4,00	8308,21
d=0,05	7525,31	0,02	197,22	172,08	158,34	4,00	8321,40
d=0	7528,71	0,02	198,51	172,13	158,60	4,00	8325,19
f=20	7498,36	0,01	147,10	172,03	131,17	3,00	8317,20
f=5	7542,73	0,02	197,10	172,00	141,43	4,00	8317,20
h=2	7411,72	0,01	147,05	172,14	221,25	3,00	8317,20
h=0.5	7600,63	0,02	247,46	171,94	110,24	5,00	8317,20
θ=0.02	7511,38	0,03	149,39	171,99	119,91	3,00	8317,20
θ=0.005	7528,41	0,01	195,05	172,03	156,62	4,00	8317,20
θ=1	6889,05	0,42	83,15	171,94	71,91	1,00	8317,20

Table 5.16: Sensitivity Comparison with Genetic Algorithm and Constant Pricing

Parameter	Profit	Decay Ratio	Inventory	Price	Holding Cost	Cycle Length	Revenue
Base Case	0,011	0,336	0,343	-0,001	-0,335	0,333	0,0002
a=200	0,004	0,490	0,518	0,000	-0,242	0,500	0,0002
a=50	0,032	0,247	0,272	-0,004	-0,257	0,250	0,0003
k=1000	0,018	0,253	0,269	-0,002	-0,333	0,250	0,0002
k=0	0,002	0,000	0,007	0,000	-0,287	0,000	0,0002
k=250	0,006	0,000	0,008	0,000	-0,405	0,000	0,0002
β=0,6	0,025	0,336	0,356	-0,004	-0,283	0,333	0,0003
β=0,15	0,005	0,336	0,350	0,000	-0,286	0,037	0,0002
C=20	0,012	0,000	0,010	-0,002	-0,467	0,000	0,0002
C=5	0,011	0,336	0,551	-0,001	-0,286	0,333	0,0002
d=0,2	0,011	0,329	0,351	0,000	-0,288	0,333	-0,9000
d=0,05	0,011	0,336	0,352	-0,002	-0,287	0,333	0,0002
d=0	0,011	0,336	0,360	-0,002	-0,286	0,333	0,0002
f=20	0,008	0,000	0,009	-0,001	-0,409	0,000	0,0002
f=5	0,014	0,342	0,352	-0,001	-0,362	0,333	0,0002
h=2	0,018	0,490	0,522	-0,003	-0,245	0,500	0,0002
h=0.5	0,006	0,253	0,266	0,000	-0,261	0,250	0,0002
θ=0.02	0,012	0,492	0,527	0,000	-0,197	0,500	0,0002
θ=0.005	0,011	0,333	12,309	-0,001	-0,286	0,333	0,0002
θ=1	0,014	0,000	0,008	0,000	-0,595	0,000	0,0002

Table 5.17: The Comparison Ratios of Genetic Algorithm Sensitivity Results with the Base Case

Parameter	Profit	Decay Ratio	Inventory	Price	Holding Cost	Cycle Length	Revenue
Base Case	7521,90	0,02	197,08	172,00	147,55	4,00	8317,20
a=200	3,25	-0,25	0,50	1,00	0,51	-0,25	3,00
a=50	-0,79	0,24	-0,38	-0,50	-0,26	0,25	-0,75
k=1000	-0,02	0,25	0,25	0,00	0,34	0,25	0,00
k=0	0,03	-0,75	-0,75	0,00	-0,65	-0,75	0,00
k=250	0,01	-0,50	-0,50	0,00	-0,41	-0,50	0,00
β=0,6	-0,55	0,00	0,00	-0,50	0,07	0,00	-0,50
β=0,15	1,11	0,00	0,00	1,00	0,08	-0,22	1,00
C=20	-0,07	-0,25	-0,25	0,00	-0,20	-0,25	0,00
C=5	0,03	0,00	0,15	0,00	0,07	0,00	0,00
d=0,2	0,00	-0,01	0,00	0,00	0,07	0,00	0,00
d=0,05	0,00	0,00	0,00	0,00	0,07	0,00	0,00
d=0	0,00	0,00	0,01	0,00	0,07	0,00	0,00
f=20	0,00	-0,25	-0,25	0,00	-0,11	-0,25	0,00
f=5	0,00	0,01	0,00	0,00	-0,04	0,00	0,00
h=2	-0,01	-0,25	-0,25	0,00	0,50	-0,25	0,00
h=0.5	0,01	0,25	0,26	0,00	-0,25	0,25	0,00
θ=0.02	0,00	0,49	-0,24	0,00	-0,19	-0,25	0,00
θ=0.005	0,00	-0,50	-0,01	0,00	0,06	0,00	0,00
θ=1	-0,08	20,01	-0,58	0,00	-0,51	-0,75	0,00

Table 5.18: Sensitivity Analysis Results with Simulated Annealing

Parameter	Profit	Decay Ratio	Inventory	Price	Holding Cost	Cycle Length	Revenue
Base Case	7473,69	0,02	197,11	172,02	139,53	4,00	8317,00
a=200	31842,87	0,01	420,68	343,50	207,81	4,20	33285,78
a=50	1538,97	0,02	177,73	86,75	162,10	7,00	2077,25
k=1000	7358,26	0,02	247,37	172,13	126,25	5,00	8317,20
k=0	7726,12	0,01	48,63	171,95	52,15	1,00	8317,20
k=250	7535,29	0,01	139,30	172,72	81,28	2,80	8317,20
β=0,6	3278,87	0,02	288,91	85,92	150,14	5,71	4158,60
β=0,15	15779,47	0,02	273,84	343,26	144,43	4,22	16634,40
C=20	6967,17	0,01	147,16	172,00	76,49	3,00	8317,20
C=5	7706,44	0,02	319,68	171,51	142,98	5,47	8317,20
d=0,2	6919,38	0,02	211,51	170,49	170,05	4,29	8308,21
d=0,05	7449,16	0,02	275,62	172,69	142,99	5,43	8321,40
d=0	7480,59	0,02	198,51	172,12	102,44	4,00	8324,89
f=20	7498,94	0,01	147,13	172,13	44,44	3,00	8317,00
f=5	7543,03	0,02	197,10	172,03	67,33	4,00	8317,00
h=2	7335,53	0,01	204,23	171,74	198,34	4,07	8317,20
h=0.5	7524,07	0,02	343,62	172,29	37,29	6,79	8317,20
θ=0.02	7420,35	0,03	199,99	171,77	102,24	3,90	8317,20
θ=0.005	7460,36	0,01	277,57	171,43	145,39	2,60	8317,20
θ=1	6345,15	0,42	83,15	171,95	37,05	1,00	8317,20

Table 5.19: Sensitivity Comparison with Simulated Annealing and Constant Pricing

Parameter	Profit	Decay Ratio	Inventory	Price	Holding Cost	Cycle Length	Revenue
Base Case	0,0048	0,336	0,34	-0,001	-0,37	0,33	0,00
a=200	0,0010	0,490	1,17	-0,001	-0,29	1,10	0,00
a=50	-0,0094	0,247	0,84	0,002	0,10	0,75	0,00
k=1000	0,0115	0,253	0,27	-0,002	-0,58	0,25	0,00
k=0	-0,0046	0,000	0,01	0,000	-0,29	0,00	0,00
k=250	-0,0030	0,000	0,44	0,004	-0,45	0,40	0,00
β=0,6	0,0023	0,336	0,99	-0,007	-0,32	0,90	0,00
β=0,15	0,0003	0,336	0,87	-0,001	-0,35	0,41	0,00
C=20	0,0050	0,000	0,01	-0,001	-0,66	0,00	0,00
C=5	0,0021	0,336	1,19	-0,004	-0,36	0,82	0,00
d=0,2	-0,0688	0,329	0,45	-0,009	-0,23	0,43	-0,90
d=0,05	0,0011	0,336	0,89	0,002	-0,36	0,81	0,00
d=0	0,0048	0,336	0,36	-0,002	-0,54	0,33	0,00
f=20	0,0082	0,000	0,01	-0,001	-0,80	0,00	0,00
f=5	0,0141	0,342	0,35	-0,001	-0,70	0,33	0,00
h=2	0,0072	0,490	1,11	-0,005	-0,32	1,04	0,00
h=0.5	-0,0036	0,253	0,76	0,002	-0,75	0,70	0,00
θ=0.02	-0,0007	0,492	1,04	-0,002	-0,32	0,95	0,00
θ=0.005	0,0022	0,333	17,94	-0,005	-0,34	-0,13	0,00
θ=1	-0,0665	0,000	0,01	0,000	-0,79	0,00	0,00

Table 5.20: Sensitivity Comparison with Genetic Algorithm Having Equally Spaced Time Intervals

Parameter	Profit	Decay Ratio	Initial Inventory	Price	Holding Cost	Cycle Length	Revenue
Base Case	7522,30	0,02	197,15	172,00	137,44	4,00	8317,30
a=200	31935,77	0,01	294,83	343,83	208,06	3,00	33286,18
a=50	1602,65	0,02	122,99	86,22	102,01	5,00	2077,28
k=1000	7406,30	0,02	247,39	172,08	184,69	5,00	8317,30
k=0	7776,24	0,01	48,65	171,95	48,57	1,00	8317,30
k=250	7604,74	0,01	97,73	171,99	81,47	2,00	8317,30
β=0,6	3353,25	0,02	196,78	86,16	147,41	4,00	4158,65
β=0,15	15860,20	0,02	197,24	343,83	147,93	3,00	16634,60
C=20	7014,03	0,01	147,20	171,99	110,10	3,00	8317,30
C=5	7776,64	0,02	226,17	172,03	147,50	4,00	8317,30
d=0,2	7515,39	0,02	196,69	171,97	146,83	4,00	8308,31
d=0,05	7525,71	0,02	197,29	172,08	147,49	4,00	8321,50
d=0	7529,11	0,02	198,58	172,13	147,73	4,00	8325,29
f=20	7498,76	0,01	147,15	172,03	122,18	3,00	8317,30
f=5	7543,14	0,02	197,17	172,00	131,74	4,00	8317,30
h=2	7412,11	0,01	147,10	172,14	206,08	3,00	8317,30
h=0.5	7601,04	0,02	247,55	171,94	102,69	5,00	8317,30
θ=0.02	7511,78	0,03	149,45	171,99	111,70	3,00	8317,30
θ=0.005	7528,81	0,01	195,12	172,03	145,88	4,00	8317,30
θ=1	6889,41	0,42	83,18	171,94	66,98	1,00	8317,30

5.2.4 Analysis of the Double Product System

When we analyze sensitivity analysis results, we obtain similar results with the single product system; hence only different parts are highlighted in this section. In double product case, most remarkable increase in profit is obtained when we double "a" and halve the parameter "β". Market share increase and price sensitivity decrease are the main reasons for the high amount of profit change. The other issue that affects objective value tremendously is "c_1" which determines the ratio of customers who chooses actual product in the case of price rise with the substitute material. Then if we increase "c_1", then more customers will choose the actual material; as a result sale of the material will rise.

In our model when we change "c_1" from 0.1 to 0.2, profit rises by 71%. On the other hand when we change from 0.1 to 0.05, profit reduces by 27% as given in Table 5.25. In Table 5.23, Table 5.26 and Table 5.22, we share the sensitivity analysis results with constant pricing, genetic and also simulated annealing algorithm simultaneously. The comparison between simulated annealing-constant pricing and genetic algorithm-constant pricing are given in Table 5.27 and Table 5.24 respectively which are found by according to the $(heuristic\,value - constant\,value)/constant\,price\,value$.

Furthermore, for the genetic algorithm base case and sensitivity results are compared in order to point out the value changes when we tune any of the parameter, which is given in Table 5.25 found according to the $(new\,parameter\,value - base\,case\,value)/base\,case\,value$. When we analyze Table 5.25, we figure our that; profit and holding costs are sensitive to all parameters,cycle length and decay amount are insensitive to "d_1", "f_1" and "g_1" parameter value change. Total inventory is insensitive to "f_1" change and average price insensitive to "d_1".

When we analyze sensitivity analysis results in Table 5.23, the change in "a_1" from 100 to 200 leads to profit increase which especially comes from the first product due to higher market potential. As a result more need arises for the first product resulting in 1.21% rise in the initial inventory with respect to the base case. This can be directly deduced from the amount revenue rise coming from the first and the second product which are 1.67% and 0.42% respectively.

When "β" is decreased, profit gain rises by 1.05%. "β" is related with price sensitivity, so when a material is less price sensitive then profit and demand values will be effected from the price fluctuations less. So revenues will also increase by 3.25% and 2.21% for the first and second product respectively.

"C_1" change is related with the order cost, hence any increase in its value will result in lower profit. "c_1" is price dependency of the first product, when it is increased then any small increment in competitors' product price will give a rise to the demand of the first product. In Table 5.25 you can see that when "'c_1" is changed from 0.1 to 0.2 profit increase is obtained by 71%, revenue for the first product is increased by 70%.

Table 5.24 gives us the comparison between constant pricing and genetic algorithm multiple pricing case. Mainly genetic algorithm outperforms constant pricing giving a rise in profit by 2.7% on average. For the base case we encounter a profit rise by 2.6%, cycle time increase by 0.5%, holding cost decrease by 26.1% and finally revenue increase by 63.4%. By using all these results, we can say that increase in revenue and decrease in holding costs are the main reasons for the profit increase.

When we examine carefully Table 5.27 and Table 5.26, we figure out that there is an increase in profit by 3%, cycle times for the first and second product rises by 77%. However decay amounts decrease by 2.97% on average. Revenue increase is directly correlated with profit gain which makes us to have 2% profit increase with the 60% revenue rise.

The comparison between genetic algorithm and simulated annealing reveals that genetic algorithm gives better results with respect to the simulated annealing and constant pricing. Profit changes from 1.8% to 4.5% in genetic, whereas in simulated profit changes from 2% to 4%. Most profit increments are achieved with the change of "a_1", "β" and "h_1" in both of the algorithms. For the base parameters profit values are as follows: 10963 for genetic algorithm, 10948.38 for simulated annealing and finally 10689 for constant pricing case.

Simulated annealing obtains a lower profit compared to genetic algorithm; for the base case its profit value is 10948, which is a 2.43% profit increase compared to constant pricing, given in Table 5.27 and also in Table 5.21. However genetic algorithm profit value is 10963 which is an increase by 2.57% with respect to the constant pricing case. Simulated annealing gives a better result when $a_1 = 50$ and $k_1 = 250$. Overall in double product case we obtain best results with genetic algorithm as in the single product system which as illustrated in Table 5.21.

Table 5.21: The Comparison of Profit Values for Double Product System

Parameter	Genetic Profit	% difference	SA Profit	% difference	Constant Profit
Base	10963,80	2,57	10948,38	2,43	10689,00
a_1=200	23554,19	1,79	23535,90	1,71	23140,00
a_1=50	5161,38	4,37	5163,23	4,41	4945,30
k_1=1000	10723,56	2,69	10705,55	2,51	10443,00
k_1=250	11047,88	2,16	11048,75	2,17	10814,00
β_1=0,6	6211,62	4,52	6193,61	4,22	5942,90
β_1=0,15	22465,12	0,52	22448,28	0,45	22348,00
C_1=20	10021,87	-2,63	10010,42	-2,75	10293,00
C_1=5	10112,96	-13,46	10095,84	-13,61	11686,00
d_1=0,2	10852,69	-3,18	10833,55	-3,35	11209,00
d_1=0,05	10974,81	-2,25	10956,52	-2,41	11227,00
c_1=0,05	8005,18	-0,20	7993,32	-0,35	8021,10
c_1=0,2	18750,46	-6,41	18731,83	-6,50	20034,00
f_1=20	10963,80	-2,29	10998,62	-1,98	11221,00
f_1=5	10986,82	-2,09	10960,75	-2,32	11221,00
h_1=0.05	11121,95	-3,00	11115,10	-3,06	11466,00
h_1=2	10862,70	-1,16	10843,61	-1,33	10990,00
θ_1=0.02	12064,88	-0,41	12043,63	-0,58	12114,00
θ_1=0.005	11932,75	-23,01	11910,43	-23,16	15500,00

Table 5.22: Double Product Sensitivity Results with Constant Pricing

Parameter	Profit	Inven. 1	Inven. 2	Cycle 1	Cycle 2	Decay 1	Decay 2	Price 1	Price 2	Holding C.	Reven.1	Reven.2
Base	10689,00	127,66	127,66	2,00	2,00	0,01	0,01	80,60	84,57	920,70	12057,30	12057,30
a_1=200	23140,00	299,19	108,08	2,00	2,00	0,01	0,01	97,64	150,80	2211,66	32214,60	17129,70
a_1=50	4945,30	53,89	149,44	2,00	2,00	0,01	0,01	72,07	51,44	1175,22	4283,37	8080,92
k_1=1000	10443,00	204,28	204,28	3,00	3,00	0,01	0,01	80,94	84,93	2085,39	18114,30	18114,30
k_1=250	10814,00	135,65	135,65	2,00	2,00	0,01	0,01	80,79	84,77	920,70	12057,30	12057,30
β_1=0,6	5942,90	109,09	149,22	2,00	2,00	0,01	0,01	62,23	48,48	588,11	7487,10	7642,80
β_1=0,15	22348,00	150,65	107,37	2,00	2,00	0,01	0,01	154,38	172,19	1050,75	25648,20	19431,90
C_1=20	10293,00	148,55	135,65	2,00	2,00	0,01	0,01	80,59	84,56	964,44	13203,00	12057,30
C_1=5	11686,00	148,55	135,65	2,00	2,00	0,01	0,01	80,59	84,56	964,44	13203,00	12057,30
d_1=0,2	11209,00	148,38	135,68	2,00	2,00	0,01	0,01	80,58	84,50	963,09	13185,90	12050,10
d_1=0,05	11227,00	148,63	135,64	2,00	2,00	0,01	0,01	80,60	84,60	965,16	13212,00	12060,90
c_1=0,05	8021,10	146,70	141,74	2,00	2,00	0,01	0,01	61,46	64,56	980,10	9943,20	9618,30
c_1=0,2	20034,00	161,41	118,98	2,00	2,00	0,01	0,01	132,85	139,28	949,50	23648,40	17417,70
f_1=20	11221,00	148,55	135,65	2,00	2,00	0,01	0,01	80,59	84,56	964,44	13203,00	12057,30
f_1=5	11221,00	148,55	135,65	2,00	2,00	0,01	0,01	80,59	84,56	964,44	13203,00	12057,30
h_1=0.05	11466,00	224,12	204,11	3,00	3,00	0,01	0,01	80,08	84,79	1118,70	19694,70	18098,10
h_1=2	10990,00	135,74	148,38	2,00	2,00	0,01	0,01	81,10	84,56	1458,63	13272,30	12064,50
θ_1=0.02	12114,00	164,24	105,21	2,00	2,00	0,01	0,01	65,96	145,99	2234,79	11827,80	16143,30
θ_1=0.005	15500,00	473,81	786,06	9,00	9,00	0,02	0,03	203,62	54,26	7798,86	105102,00	43311,60

Table 5.23: Double Product Sensitivity Analysis Results with Genetic Algorithm

Parameter	Profit	Inven. 1	Inven. 2	Cycle 1	Cycle 2	Decay 1	Decay 2	Price 1	Price 2	Holding C.	Reven.1	Reven.2
Base	10963,80	125,41	125,79	3,00	3,00	0,01	0,01	89,87	89,49	680,09	19701,94	19701,94
a_1=200	23554,19	276,87	100,26	3,00	3,00	0,01	0,01	108,82	159,42	1952,22	52670,44	27925,21
a_1=50	5161,38	84,17	232,03	5,00	5,00	0,01	0,01	79,83	55,26	610,47	11598,20	22189,30
k_1=1000	10723,56	253,57	255,19	6,00	6,00	0,02	0,02	90,60	89,34	2657,58	39560,63	38890,54
k_1=250	11047,88	125,41	125,79	3,00	3,00	0,01	0,01	89,87	89,26	678,10	19701,94	19603,00
β_1=0,6	6211,62	101,37	137,62	3,00	3,00	0,01	0,01	68,83	51,65	360,53	12183,05	12457,36
β_1=0,15	22465,12	282,18	202,12	6,00	6,00	0,02	0,02	172,48	182,26	2894,82	83813,84	63265,47
C_1=20	10021,87	210,53	211,66	5,00	5,00	0,01	0,01	90,35	89,34	1808,16	32922,46	32638,35
C_1=5	10112,96	125,41	125,80	3,00	3,00	0,01	0,01	89,87	89,24	655,06	19701,94	19600,06
d_1=0,2	10852,69	209,87	211,35	5,00	5,00	0,01	0,01	90,31	89,27	1791,87	32805,88	32589,37
d_1=0,05	10974,81	125,52	125,77	3,00	3,00	0,01	0,01	89,88	89,26	676,52	19723,50	19611,81
c_1=0,05	8005,18	126,56	131,44	3,00	3,00	0,01	0,01	68,58	68,03	698,16	15173,95	15606,96
c_1=0,2	18750,46	129,51	110,35	3,00	3,00	0,01	0,01	147,98	147,34	643,99	33505,35	28385,65
f_1=20	10963,80	125,41	125,79	3,00	3,00	0,01	0,01	89,87	89,22	679,76	19701,94	19701,94
f_1=5	10986,82	125,80	125,40	3,00	3,00	0,01	0,01	89,25	89,87	669,82	19599,08	19701,94
f_1=0.05	11121,95	168,60	167,79	4,00	4,00	0,01	0,01	89,13	90,11	696,65	26095,20	26299,95
h_1=2	10862,70	125,06	125,92	3,00	3,00	0,01	0,01	90,72	89,23	956,30	198289,07	19616,71
θ_1=0.02	12064,88	143,03	96,83	3,00	3,00	0,02	0,01	73,35	155,95	2229,37	18049,26	26409,68
θ_1=0.005	11932,75	199,55	482,65	9,00	9,00	0,01	0,02	226,49	57,53	4275,40	78546,20	47145,16

Table 5.24: Comparison of Genetic Algorithm Results with Constant Pricing Case in Double Product System

Parameter	Profit	Inven. 1	Inven. 2	Cycle 1	Cycle 2	Decay 1	Decay 2	Price 1	Price 2	Holding C.	Reven.1	Reven.2
Base	0,026	0,599	-0,015	0,500	0,500	0,162	0,162	0,115	0,055	-0,261	0,634	0,634
a_1=200	0,018	0,506	-0,072	0,500	0,500	0,162	0,162	0,114	0,054	-0,117	0,635	0,630
a_1=50	0,044	1,542	0,553	1,500	1,500	0,494	0,497	0,108	0,069	-0,481	1,708	1,746
k_1=1000	0,027	1,020	0,249	1,000	1,000	0,374	0,374	0,119	0,049	0,274	1,184	1,147
k_1=250	0,022	0,505	-0,073	0,500	0,500	0,162	0,162	0,112	0,050	-0,263	0,634	0,626
β_1=0,6	0,045	0,512	-0,078	0,500	0,500	0,162	0,162	0,106	0,061	-0,387	0,627	0,630
β_1=0,15	0,005	2,049	0,883	2,000	2,000	0,580	0,580	0,117	0,055	1,755	2,268	2,256
C_1=20	-0,026	1,307	0,560	1,500	1,500	0,497	0,497	0,121	0,054	0,875	1,494	1,707
C_1=5	-0,135	1,307	-0,073	0,500	1,500	0,162	0,162	0,115	0,052	-0,321	0,492	0,626
d_1=0,2	-0,032	1,302	0,558	1,500	1,500	0,494	0,497	0,121	0,053	0,861	1,488	1,704
d_1=0,05	-0,022	0,375	-0,073	0,500	0,500	0,162	0,162	0,115	0,052	-0,299	0,493	0,626
c_1=0,05	-0,002	0,404	-0,073	0,500	0,500	0,162	0,162	0,116	0,051	-0,288	0,526	0,623
c_1=0,2	-0,064	0,306	-0,073	0,500	0,500	0,162	0,162	0,114	0,055	-0,322	0,417	0,630
f_1=20	-0,023	0,374	-0,073	0,500	0,500	0,162	0,162	0,115	0,052	-0,295	0,492	0,634
f_1=5	-0,021	0,378	-0,076	0,500	0,500	0,162	0,162	0,107	0,059	-0,305	0,484	0,634
h_1=0.05	-0,030	0,224	-0,178	0,333	0,333	0,065	0,060	0,113	0,059	-0,377	0,325	0,453
h_1=2	-0,012	0,500	-0,151	0,500	0,500	0,162	0,162	0,119	0,052	-0,344	13,940	0,626
θ_1=0.02	-0,004	0,417	-0,080	0,500	0,500	0,163	0,162	0,112	0,064	-0,002	0,526	0,636
θ_1=0.005	-0,230	-0,314	-0,386	0,000	0,000	-0,249	-0,249	0,112	0,057	-0,452	-0,253	0,089

Table 5.25: The Comparison Ratios of Genetic Algorithm Sensitivity Results with the Base Case

Parameter	Profit	Inven. 1	Inven. 2	Cycle 1	Cycle 2	Decay 1	Decay 2	Price 1	Price 2	Holding C.	Reven.1	Reven.2
Base	10958,40	178,88	179,43	3,00	3,00	0,01	0,01	84,80	84,44	813,45	19905,42	19905,42
a_1=200	1,15	1,21	-0,20	0,00	0,00	0,00	0,00	0,21	0,78	1,87	1,67	0,42
a_1=50	-0,53	-0,33	0,84	0,67	0,67	0,66	0,66	-0,11	-0,38	-0,10	-0,41	0,13
k_1=1000	-0,02	1,02	1,03	1,00	1,00	0,99	0,99	0,01	0,00	2,91	1,01	0,97
k_1=250	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	-0,01
β_1=0,6	-0,43	-0,19	0,09	0,00	0,00	0,00	0,00	-0,23	-0,42	-0,47	-0,38	-0,37
β_1=0,15	1,05	1,25	0,61	1,00	1,00	0,99	0,99	0,92	1,04	3,26	3,25	2,21
C_1=20	-0,09	0,68	0,68	0,67	0,67	0,66	0,66	0,01	0,00	1,66	0,67	0,66
C_1=5	-0,08	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	-0,04	0,00	-0,01
d_1=0,2	-0,01	0,67	0,68	0,67	0,67	0,66	0,66	0,00	0,00	1,63	0,67	0,65
d_1=0,05	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	-0,01	0,00	0,00
c_1=0,05	-0,27	0,01	0,04	0,00	0,00	0,00	0,00	-0,24	-0,24	0,03	-0,23	-0,21
c_1=0,2	0,71	0,03	-0,12	0,00	0,00	0,00	0,00	0,65	0,65	-0,05	0,70	0,44
f_1=20	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
f_1=5	0,00	0,00	0,00	0,00	0,00	0,00	0,00	-0,01	0,00	-0,02	-0,01	0,00
h_1=0.05	0,01	0,34	0,33	0,33	0,33	0,34	0,33	-0,01	0,01	0,02	0,32	0,33
h_1=2	-0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,00	0,41	9,06	0,00
θ_1=0.02	0,10	0,14	-0,23	0,00	0,00	0,99	0,00	-0,18	0,74	2,28	-0,08	0,34
θ_1=0.005	0,09	0,59	2,84	2,00	2,00	0,50	1,97	1,52	-0,36	5,29	2,99	1,39

Table 5.26: Double Product Sensitivity Analysis Results with Simulated Annealing Algorithm

Parameter	Profit	Inven. 1	Inven. 2	Cycle 1	Cycle 2	Decay 1	Decay 2	Price 1	Price 2	Holding C.	Reven.1	Reven.2
Base	10948,38	125,41	125,79	2,63	2,63	0,00	0,00	89,87	89,50	6797,10	19276,21	19276,21
a_1=200	23535,90	276,87	100,26	3,00	3,00	0,03	0,05	108,55	159,15	1950,41	49050,55	24924,00
a_1=50	5163,23	74,54	205,30	4,44	4,44	0,06	0,04	79,86	55,27	4622,67	11047,47	21629,98
k_1=1000	10705,55	224,37	225,80	5,33	5,33	0,05	0,05	90,47	89,47	11293,76	39560,63	38890,54
k_1=250	11048,75	125,41	125,79	2,50	2,50	0,03	0,03	89,85	89,25	6746,92	19207,23	19110,77
β_1=0,6	6193,61	101,37	137,62	2,67	2,67	0,02	0,02	68,83	51,65	360,87	12179,39	12476,27
β_1=0,15	22448,28	282,18	202,12	6,00	6,00	0,01	0,01	172,66	182,08	1655,95	83715,11	63167,24
C_1=20	10010,42	253,78	255,15	6,00	6,00	0,01	0,01	90,35	89,34	1430,19	31931,44	31655,89
C_1=5	10095,84	125,41	125,80	3,00	3,00	0,03	0,03	89,87	89,24	355,94	19659,50	19557,83
d_1=0,2	10833,55	209,87	211,35	5,00	5,00	0,06	0,01	90,31	89,27	968,35	32723,15	32507,40
d_1=0,05	10956,52	125,52	125,77	3,00	3,00	0,03	0,03	89,88	89,26	676,15	19681,11	19654,02
c_1=0,05	7993,32	126,56	131,44	3,00	3,00	0,02	0,02	68,58	68,04	379,80	14763,15	15169,39
c_1=0,2	18731,83	110,51	94,15	3,00	3,00	0,01	0,01	148,19	147,23	351,19	33447,28	28343,87
f_1=20	10998,62	71,51	71,73	1,75	0,00	0,02	0,02	89,84	89,70	238,81	19453,10	19453,09
f_1=5	10960,75	147,62	147,14	3,50	3,50	0,04	0,04	88,81	89,63	499,22	19106,95	19207,23
h_1=0.05	11115,10	129,95	129,32	3,11	3,11	0,03	0,03	88,88	89,98	225,96	26051,39	26255,80
h_1=2	10843,61	125,06	125,92	3,00	3,00	0,03	0,03	90,71	89,23	519,64	20543,40	19574,45
θ_1=0.02	12043,63	143,03	96,83	3,00	3,00	0,03	0,00	73,33	155,95	1305,08	18044,73	26343,36
θ_1=0.005	11910,43	199,55	48,27	9,00	9,00	0,01	0,01	226,55	54,94	2675,45	77394,52	46440,45

Table 5.27: Comparison of Simulated Annealing Results with Constant Pricing Case in Double Product System

Parameter	Profit	Inven. 1	Inven. 2	Cycle 1	Cycle 2	Decay 1	Decay 2	Price 1	Price 2	Holding C.	Reven.1	Reven.2
Base	0,02	-0,02	-0,01	0,31	0,31	-0,57	-0,57	0,12	0,06	6,38	0,60	0,60
a_1=200	0,02	-0,07	-0,07	0,50	0,50	3,17	6,79	0,11	0,06	-0,12	0,52	0,46
a_1=50	0,04	0,38	0,37	1,22	1,22	6,95	4,07	0,11	0,07	2,93	1,58	1,68
k_1=1000	0,03	0,10	0,11	0,78	0,78	3,70	3,70	0,12	0,05	4,42	1,18	1,15
k_1=250	0,02	-0,08	-0,07	0,25	0,25	3,69	3,69	0,11	0,05	6,33	0,59	0,58
β_1=0,6	0,04	-0,07	-0,08	0,33	0,33	2,14	2,12	0,11	0,07	-0,39	0,63	0,63
$beta_1$=0,15	0,00	0,87	0,88	2,00	2,00	0,59	1,02	0,12	0,06	0,58	2,26	2,25
C_1=20	-0,03	0,71	0,88	2,00	2,00	-0,03	-0,03	0,12	0,06	0,48	1,42	1,63
C_1=5	-0,14	-0,16	-0,07	0,50	0,50	3,34	3,34	0,12	0,06	-0,63	0,49	0,62
d_1=0,2	-0,03	0,41	0,56	1,50	1,50	7,11	-0,19	0,12	0,06	0,01	1,48	1,70
d_1=0,05	-0,02	-0,16	-0,07	0,50	0,50	3,33	3,35	0,12	0,06	-0,30	0,49	0,63
c_1=0,05	0,00	-0,14	-0,07	0,50	0,50	1,38	2,48	0,12	0,05	-0,61	0,48	0,58
c_1=0,2	-0,06	-0,32	-0,21	0,50	0,50	0,30	-0,20	0,12	0,06	-0,63	0,41	0,63
f_1=20	-0,02	-0,52	-0,47	-0,13	-1,00	1,55	1,55	0,11	0,06	-0,75	0,47	0,61
f_1=5	-0,02	-0,01	0,08	0,75	0,75	4,05	4,05	0,10	0,06	-0,48	0,45	0,59
h_1=0.05	-0,03	-0,42	-0,37	0,04	0,04	2,03	2,02	0,11	0,06	-0,80	0,32	0,45
h_1=2	-0,01	-0,08	-0,15	0,50	0,50	3,34	3,34	0,12	0,06	-0,64	0,55	0,62
θ_1=0.02	-0,01	-0,13	-0,08	0,50	0,50	1,13	-0,51	0,11	0,07	-0,42	0,53	0,63
θ_1=0.005	-0,23	-0,58	-0,94	0,00	0,00	-0,30	-0,61	0,11	0,01	-0,66	-0,26	0,07

Chapter 6

CONCLUSIONS

In this study we analyze the problem of jointly determining the profit maximizing pricing and lot sizing decisions with intertemporal price discrimination in a deterministic setting in which demand not only depends on the price but also on the freshness of the products. Furthermore, it is also assumed that products decay at a certain rate which increases the complexity of the problem. Besides providing evidence for the benefits of the dynamic pricing, we especially show its impact on operational costs (order and holding cost) and also order cycle decisions.

We explicitly find results for the optimal order size, optimal number of times the prices should be changed and the optimal price values at each time for the single product and double product cases. Heuristic algorithms are proposed in order to find best profit giving time values. Throughout the experiments we compare the results of dynamic pricing with constant pricing case. When we take price change cost zero, then we obtain continuous pricing where both continuous and constant pricing cases are the extreme points that we test. In numerical experiments we show the effect of parameter change on the profit, order and holding costs, revenues and also other parameters. When we concern about lot size, we observe that optimal lot size increases with increasing holding cost. Same effect is also observed with the cycle length which increases with the fixed cost and decreases with the holding cost and decay coefficient value.

Our model can be extended in several ways. Firstly, different types of demand functions or decay processes can be analyzed in detail and explicit results can be obtained. In addition, this problem can be analyzed in the stochastic setting in which the demand and also the decaying process is random. Even though our results can form a basis for developing coordinated pricing and inventory policies that can be also applied in stochastic environments, the performances of such policies need to be investigated and effective policies need to be developed for stochastic systems.

BIBLIOGRAPHY

[1] P. L. Abad, *Optimal pricing and lot sizing under conditions of perishability and partial backordering*, Management Science, 1996, vol.42, pp. 1093–1104.

[2] P. L. Abad,*Optimal pricing and lot sizing under conditions of perishability and partial backordering and lost sale*,European Journal of Operational Research, 2003,vol.144,pp. 677–685.

[3] P. L. Abad,*Optimal price and order size for a reseller under partial backordering*,Computers and Operations Research, 2001,vol.28,pp. 53–65.

[4] P. L. Abad,*Optimal policy for a reseller when the supplier offers a temporary reduction in price* , Decision Sciences, 1997, vol.28,pp. 637.

[5] Abdulhakim A.Al-babtain,*Necessary Condition for the Optimality on Multi-Item Inventory Model*,International Mathematical Forum,2010,no.2, pp.89-98.

[6] Lakdere Benkherouf, *On an inventory model with deteriorating items and decreasing time-varying demand and shortages*, European Journal of Operational Research,1995,vol. 86,pp. 293–299.

[7] D.K Bhattacharya, *Production manufacturing and logistics on multi item inventory*,European Journal of Operational Research,2005,vol. 162, pp. 786–791.

[8] Jhuma Bhowmick, G.P Samanta, *A Deterministic inventory model of deterioratingn items with two Rates of production, shortages, and variable production cycle*,International Scholarly Research Network, ISRN Applied Mathematics,2011.

[9] A.K. Bhunia, M.Maiti, *Deterministic inventory model for deteriorating items with finite rate of replenishment dependent on inventory level* Computers Operations Research, 1998, vol. 25, no. 11, pp. 997–1006.

[10] Myers Bret, *Analysis of an inventory system with product perishability and substitution: A simulation-optimization approach*, Doctor of Philosophy Thesis, 2009, Drexel University.

[11] Burwell Timothy H., Dave S.Dinesh, Fitzpatrick Kathy E., Melvin R., *An Inventory Model with Planned Shortages and Price Dependent Demand*, Decision Sciences, 1991, vol.22, pp.1187.

[12] Covert, R. P., Philip, G. C., *An EOQ model with Weibull deterioration*, AIIE Transactions, 1973, vol. 5, pp. 323–326.

[13] Deniz Borga, *Essays on perishable inventory management*, Doctor of Philosophy Thesis, Carnegie Mellon University, 2007.

[14] P-S You, *Inventory policy for products with price and time-dependent demands*,2005, Journal of the Operational Research Society, vol.56, pp.870–873.

[15] DaeSoo Kim, Won J. Lee, *Optimal joint pricing and lot sizing with fixed and variable capacity*, 1998, European Journal of Operational Research, vol.109, pp. 212–227.

[16] Transchel Sandra, Minner Stefan, *The impact of dynamic pricing on the economic order decision*, European Journal of Operational Research,2009, vol. 198, pp. 773–789.

[17] Chew Peng Ek, Lee Chulung, *Joint inventory allocation and pricing decisions for perishable products*, International Journal of Production Economics, 2009, vol.120, pp. 139–150.

[18] Ghare, P. M. and Schrader, G. P., *A Model for an exponentially inventory*, Journal of Industrial Engineering,1963, vol. 14, pp. 238–243.

[19] Gupta Diwakar, Hill Arthur, Chameeva Tatiana Bouzdine, *A pricing model for clearing end-of-season retail inventory*, European Journal of Operational Research, 2006, vol. 170, pp. 518–540.

[20] Pekelman Dov, *Simultaneous price production decisions*, Center for Mathematical Studies in Business and Economics, University of Chicago, 1971.

[21] Drezner Zvi, Gurnani Haresh, Pasternack A. Barry, *An EOQ model with substitutions between products*,Journal of the Operational Research Society, 1995, vol. 46, pp. 887-891.

[22] Burke EK, Bai R., Kendall G., *Heuristic,meta-heuristic and hyper-heuristic approaches for fresh produce inventory control and shelf space allocation*, Journal of the Operational Research Society, 2008, vol. 59, pp. 1387–1397.

[23] Rafaat Fred, *Survey of literature on continuously deteriorating inventory models*, The Journal of the Operational Research Society, 1991, vol. 42, pp. 22–37.

[24] Bitran Gabriel, Caldentey Rene, Vialz Raimundo, *Pricing policies for perishable products with demand substitution*,Manufacturing and Service Operations Management,vol. 5, pp. 203–229.

[25] Widyadana Gede Agus, Wee Wing Hui, *Supply chain retailer price markdown policy for deteriorating inventory.*

[26] Giri B. C., Chaudhuri K. S., *Deterministic models of perishable inventory with stock-dependent demand rate and nonlinear holding cost*,1998, European Journal of Operational Research, vol. 105, pp. 467–474.

[27] Goyal S. K., Giri B.C., *Recent trends in modeling of deteriorating inventory*, European Journal of Operational Research, 2001, vol. 134, pp. 1–16.

[28] Lee Hau L., Rosenblatt J. Meir, *A Generalized quantity discount pricing model to increase supplier's profit*, Management Science, 1986, vol. 32,no.9.

[29] Deng Shiming, Yano A. Candace, *Joint production and pricing decisions with setup costs and capacity constraints*, Management Science, 2006, vol. 52, pp. 741–756.

[30] Wee Hui Ming, *Deteriorating inventory model with quantity discount, pricing and partial backordering*, International Journal of Production Economics, 1999, vol.59, pp. 511–519.

[31] Eliashberg Jehoshua, Steinberg Richard, *Marketing and production decisions in an industrial channel of distribution*, Management Science, 1987, vol. 33, no. 8.

[32] Thomas Joseph, *Price production decisions with deterministic demand*, Management Science,1970, vol.16, no. 11.

[33] Rajaram Kumar, Tang S. Christopher, *The impact of product substitution on retail merchandising*, European Journal of Operational Research, 2001, vol.135, pp. 582–601.

[34] Liu Rujing, *Joint Pricing and ordering decisions for perishable products*, Degree of Doctor Philosophy Thesis, 2006, National University of Singapore.

[35] Mukhopadhyaya S., Mukherjeea R.N., Chaudhuri K.S., *Joint pricing and ordering policy for a deteriorating inventory*, Computers and Industrial Engineering, 2004, vol.47, pp. 339–349.

[36] Sana Shib Sankar, *Optimal selling price and lotsize with time varying deterioration and partial backlogging*, Applied Mathematics and Computation,2010, vol. 217, pp. 185–194.

[37] Mishra S. S., Mishra P.P., *Price determination for an EOQ model for deteriorating items under perfect competition*, Computers and Mathematics with Applications, 2008, vol. 56, pp. 1082–1101.

[38] Bakker Monique, Riezebos Jan, Teunter H. Ruud, *Review of inventory systems with deterioration since 2001*, European Journal of Operational Research, 2012, vol. 221, pp. 275–284.

[39] Fleischmann Moritz, Hall Joseph M., Pyke David F., *A Dynamic pricing model for coordinated sales and operations*, Erasmus Research Institute of Management, 2005.

[40] Yinping Mu, Jingming Pan, *Integration of pricing and production decisions for substitutable multiple products under uncertainty.*

[41] Jeyanthi N., Radhakrishnan P., *Optimizing multi product inventory using genetic algorithm for efficient supply chain management involving lead time*, International Journal of Computer Science and Network Security, 2010, vol.10, no.5.

[42] Nagare Madhukar, Dutta Pankaj, *Continuous review model for perishable products with inventory dependent demand*, Proceedings of the Multi Conference of Engineers and Computer Scientists, 2012, Hong Kong.

[43] Nahmias Steven, *Perishable Inventory Theory: A Review*, 1982.

[44] Netessine Serguei, *Dynamic pricing of inventory/capacity with infrequent price changes*, European Journal of Operational Research, 2006, vol. 174, pp. 553–580.

[45] Panda S., Saha S., Basu M., *An EOQ model for perishable products with discounted selling price and stock dependent demand*, CEJOR, 2009.

[46] Porteus Evan L., *Investing in reduced setups in the EOQ model*, Management Science, 1985, vol. 31, no. 8.

[47] Rakesh Arvind Rajan, Steinberg Richard, *Dynamic pricing and ordering decisions by a monopolist*, Management Science, 1992, vol. 38, no.2, pp. 240–262.

[48] Singh Trailokyanath, Sahu Sudhir Kumar, *An inventory model for deteriorating items with different constant demand rates*, African Journal of Mathematics and Computer Science Research, 2012, vol. 5, pp. 158–168.

[49] Singh Trailokyanath, Pattnayak Hadibandhu, *An EOQ Model for deteriorating items with linear demand, variable deterioration and partial backlogging*,Journal of Service Science and Management, 2013, vol. 6, pp. 186–190.

[50] Jorgensen Steffen, Kort M. Peter, *Optimal pricing and inventory policies: centralized and decentralized decision making*, European Journal of Operational Research, 2002, vol. 138, pp. 578–600.

[51] Transchel Sandra, Minner Stefan, *Coordinated Lot-sizing and Dynamic Pricing under a Supplier All-units Quantity Discount*, Verband der Hochschullehrer für Betriebswirtschaft, 2008, vol. 1, pp. 125–141.

[52] Dalfard Vahid Majazi, Nosratian Nasim Ekram, *A new pricing constrained single-product inventory-production model in perishable food for maximizing the total profit*, Neural Computations and Applications, 2012.

[53] Mishra Vinod Kumar, Singh Sahab Lal, *Deteriorating Inventory Model with Time Dependent Demand and Partial Backlogging*, Applied Mathematical Sciences, 2010, vol. 4, no. 72,pp. 3611–3619.

[54] Wagner Harvey M., Whitin M. Thomson, *Dynamic version of the economic lot size model*,Supplement, December 2004,vol 50,pp. 1770-1774.

[55] Wee Hui Ming, *A replenishment policy for items with a price-dependent demand and a varying rate of deterioration*,Production and Planning Control, 1997, vol. 8, no. 5, pp. 494– 499.

[56] Pan Wei, *Determining Optimal Selling Price, Order Size and the Number of Price Changes with Weibull Distribution Deterioration*, Journal of Computers, 2009, vol. 4, no. 12.

[57] Whitin T. M., *Inventory control and price theory*,Management Science, 1955,vol. 2,pp. 61-68

[58] Yadavalli V.S.S., C. van Schoor,Strasheim J.J., Udayabaskaran S., *A single product perishing inventory model with demand interaction*, Orion,2004, pp. 109–124.

[59] Yadavalli V.S.S.,Sivakumar B., Arivarignan G., Adetunji O., *A multi-server perishable inventory system with negative customer*, Computers and Industrial Engineering, 2011, vol.61,pp. 254–273.

VITA

Aylin polat was born in Gaziantep, Turkey, on March 25, 1985. She received her B.Sc. in Chemical Engineering and Food Engineering from Middle East Technical University, Ankara, in 2010. In September 2010, she started her M.Sc. degree at Koç University. Until July 2012, she worked as a teaching assistant. Her research focus was on Inventory Management and Heuristics.