Optimization of a Production Inventory Model with Reliability Considerations

MD. ABDULLA AL MASUD

DEPARTMENT OF INDUSTRIAL AND PRODUCTION ENGINEERING
BANGLADESH UNIVERSITY OF ENGINEERING & TECHNOLOGY
DHAKA-1000, BANGLADESH

JANUARY, 2012

Optimization of a Production Inventory Model with Reliability Considerations

BY
MD. ABDULLA AL MASUD

A thesis submitted to the Department of Industrial and Production Engineering, Bangladesh University of Engineering & Technology, in partial fulfillment of the requirements for the degree of Master of Science in Industrial and Production Engineering.

DEPARTMENT OF INDUSTRIAL AND PRODUCTION ENGINEERING
BANGLADESH UNIVERSITY OF ENGINEERING & TECHNOLOGY
DHAKA-1000, BANGLADESH

JANUARY, 2012

CERTIFICATION OF APPROVAL

The thesis titled "**Optimization of a Production Inventory Model with Reliability Considerations**" submitted by **Md. Abdulla Al Masud**, Student no.: **1009082006**, has been accepted as satisfactory in partial fulfillment of the requirement of the degree of **Master of Science in Industrial and Production Engineering** on January, 2012.

Dr. Abdullahil Azeem
Thesis Supervisor
Professor & Head
Department of Industrial and Production Engineering
Bangladesh University of Engineering and Technology
Dhaka-1000, Bangladesh

CANDIDATE'S DECLARATION

It is hereby declared that this thesis or any part of it has not been submitted elsewhere for the award of any degree or diploma

Md. Abdulla Al Masud

To the Almighty

To my parents

ACKNOWLEDGEMENT

The author expresses his sincere gratitude and profound indebtness to Dr. Abdullahil Azeem, Professor, Department of Industrial & Production Engineering, BUET, Dhaka-1000, under whose continuous supervision this thesis was carried out. His affectionate guidance, valuable suggestions and inspirations at every stage made this work possible.

I would like to thanks Dr. A. K. M. Masud, Professor, Department of Industrial & Production Engineering, BUET, Dhaka-1000, for his inspiration, motivation and kind support regarding this thesis.

I would like to extend my gratitude to Mr. Sanjoy Kumar Paul and Mr. Sayed Mithun Ali, both are Assistant Professor, Department of Industrial & Production Engineering, BUET, Dhaka-1000 for helping me to enrich this work with necessary suggestions.

I would like to express my thanks and gratitude to all the members of Department of Industrial & Production Engineering, BUET, Dhaka-1000 for their kind cooperation and valuable suggestions to complete the thesis timely.

Finally, I would like to extend my sincere and thanks to my parents for their continuous inspiration and support that encouraged me to complete the thesis successfully.

ABSTRACT

The classical production inventory control models assume that products are produced by perfectly reliable production process with no defective items. However, in reality, products are not always perfect but are directly affected by the reliability of the production process. While the reliability of the production process cannot be increased without a price, its rejection and inspection cost can be reduced with investment in flexibility and reliability improvement. In this thesis, a production inventory model with reliability of production process consideration is developed which considers the combined effect of production cost, setup cost, holding cost, inspection cost, depreciation cost, rejection cost and backorder cost on total cost minimization. The economic production lot size and the reliability of the production process along with the production period are the decision variables and total cost per cycle is the objective function which is to be minimized. A meta-heuristic Particle Swarm Optimization (PSO) algorithm is used to solve the unconstrained non integer non linear form of objective function as it can generate accurate result with a shorter computational time with stable convergence. Some numerical examples have been presented to explain the model. The results obtained from PSO algorithm is compared with the results of Genetic Algorithm (GA) applying on the same inventory model and found satisfactory.

TABLE OF CONTENT

LIST OF TABLES

Table **Page no.**

LIST OF FIGURES

NOMENCLATURE

C= Per unit production cost

D = Demand rate

P= Production rate

r= Reliability of the production process (a decision variable)

T= Cycle time (a decision variable)

Q= Lot size in no of units per cycle (a decision variable)

h= Holding cost per unit

S= Setup cost per Cycle

I= Inspection cost per unit

l= positive constants

m= positive constants

n= positive constant

J=Rejection cost per unit

Y= Optimal Backorder quantity

α= Backorder administrative cost per unit

β= Backorder cost per unit per year

CHAPTER 1
INTRODUCTION

Inventory control is the supervision of supply, storage and accessibility of items in order to ensure an adequate supply without excessive oversupply. It can also be referred as the process of managing the timing and the quantities of goods to be ordered and stocked, so that demands can be met satisfactorily and economically. Inventory control policies are decision rules that focus on the trade-off between the costs and benefits of alternative solutions to questions of when and how much to order for each different type of item. Success of inventory control depends on some important issues i.e. uncertainty about the size of future demands, uncertainty of inventory cost, uncertainty of lead time, reliability of the production process etc. The organizations and managers are most of the times interested in and worried for inventory costs. The control of these costs of the past, present and future is part of the job of all the managers in a company. In the companies that try to have profits, the control of costs affects directly to them. Knowing the costs of inventory is essential for decision making regarding price and mix assignation of products and services.

1.1 Rationale of the Study

Inventories are used to serve a variety of functions in a company such as coordinating operations, smoothing production, achieving economies of scale, improving customer service etc. But keeping a high level of inventory is a costly exercise and thus no surprise to find that many managers generally regard inventories as necessary evils. The economic production quantity (EPQ) model is used in manufacturing environments to assist firms in determining the optimal production lot size that minimizes the overall production inventory costs. For this reason, EPQ model has been widely used for more than three decades as an important tool to control the inventory and it is the most powerful tool to help practitioners and engineer to make a decision. In this regard, a new generalized EPQ model is proposed considering production cost, holding cost, setup cost, inspection cost, depreciation and insurance cost, defective units cost and backorder cost along with reliability of the production process.

1.2 Objectives

The objectives of the proposed thesis work are:

i. To develop a production inventory model considering production cost, holding cost, setup cost, inspection cost, depreciation and interest cost, defective units cost and backorder cost along with reliability or imperfect production process consideration.
ii. To optimize the production inventory model by minimizing total cost using Particle Swarm Optimization (PSO) algorithm.
iii. To compare the results of PSO algorithm with Genetic Algorithm (GA), applying it on the same inventory model.

This thesis, however, will give possible clues in the development of production inventory model by providing numerical results to help on understanding, formulation and analysis of such inventory model.

1.3 Methodology

The research work is theoretical in nature. A production inventory model is developed incorporating cost of quality (depreciation, inspection and rejection costs) and backorder cost with traditional EPQ model. The model considers the effect of reliability of the production process on all cost items. The production inventory model is formed as unconstrained non integer non linear form which is optimized to determine the numerical values of different decision variables (lot size per cycle, reliability of the production process and duration cycle time). The methodology would be as follows:

i. Equations for production cost, holding cost, setup cost, depreciation & interest cost, inspection cost and rejection cost considering reliability has been obtained and developed for the new EPQ model.

ii. Two types of backorder cost are considered: administrative backorder cost and backorder cost due to loss of goodwill. Mathematical equation for total backorder cost is developed.

iii. A new cost function has been developed considering production cost, setup cost and holding cost along with inspection cost, depreciation and insurance cost, rejection cost and backorder cost

iv. Total cost of production, economic lot size, cycle time and process reliability will be determined and optimized using PSO algorithm.

v. The model will be explained with a numerical illustration.

vi. The results obtained from PSO will be compared with the results derived from GA.

CHAPTER 2
LITERATURE REVIEW

Lack of synchronization in the production system, along with inherent uncertainty in material supply and demand, makes holding inventory a necessity, yet keeping a high level of inventory is a costly exercise. It is thus no surprise to find that many managers generally regard inventories as necessary evils. Representing a significant portion of a company's assets, inventories are used to serve a variety of functions, chief among which are: (1) coordinating operations, (2) smoothing production, (3) achieving economies of scale and (4) improving customer service. The determination of the most cost effective production quantity under rather stable conditions is commonly known as the classical economic production quantity (EPQ). The economic production quantity (EPQ) model is used in manufacturing environments to assist firms in determining the optimal production lot size that minimizes the overall production – inventory costs. For this reason, EPQ model has been widely used for more than three decades as an important tool to control the inventory and it is the most powerful tool to help practitioners and engineer to make a decision.

Though EPQ model is a powerful tool, it did not represent the real world problems in some situations. Regardless of such an acceptance, the analysis for finding an economic production quantity has several weaknesses. The obvious is the number of unrealistic assumptions which lead many researchers to make extensions in several aspects of the original EPQ model. Here, first various previous researches in EPQ model will be discussed, the scope of future research will be projected and finally objective of this thesis will be outlined.

Salameh et al. (2000) [1] was the first, who hypothesizes a production/inventory situation where items received or produced are not of perfect quality. Items of imperfect quality could be used in another production/inventory situation such as less restrictive process and acceptance control. This paper extends the traditional EPQ/EOQ model by accounting for imperfect quality items when using the EPQ/EOQ formulae. Their paper also considers the issue that poor-quality items are sold as a single batch by the end of the

100% screening process. They consider a case where the lot delivered instantaneously with a purchasing price and an ordering cost. It is assumed that each lot received contains some percentage defectives with a known probability density function. Unlike the assumption of defective items can be reworked instantaneously at a cost, in this paper it is assumed that defective items are sold as a single batch at a discounted price. A 100% percent screening process of the lot is conducted. Items of poor quality are kept in stock and sold prior to receiving the next shipment as a single batch. The optimum operating inventory doctrine is obtained by trading of total revenues per unit time, procurement cost per unit time, the inventory carrying cost per unit time and item screening cost per unit time so that their sum will be a maximum.

Goyal et al. (2002) [2] present a simple approach for determining the economic production quantity for an item with imperfect quality developed by Salameh [1]. They compare the results based on the simple approach with the optimal method for determining the lot size and show that almost optimal results are obtained using the simple approach.

In many cases, large piles of consumer goods are often associated with on sale items to induce more sales and profits. Teng and Chang (2005) [3] therefore assumed that the demand is a function of the selling price and the stock on display. With perishable goods, shoppers usually walk away if they cannot find the product they want. Therefore, to avoid lost sales, shortages were not allowed in this paper. As too many goods piled up in everyone's way leaves a negative impression on buyers, they imposed a limited maximum number of stock items without leaving a negative impression on buyers. It may be desirable to order large quantities, resulting in stock remaining at the end of the cycle, due to the potential profits resulting from the increased demand. Consequently, a terminal zero-inventory condition was imposed at the end of the replenishment cycle. In summary, a single-item deterministic inventory model for deteriorating items with constant production rate is presented in their paper. In addition, they impose a ceiling on the number of on-display stocks because too much stock leaves a negative impression on the buyer and the amount of shelf space is limited. Finally, sensitivity analysis is applied on the parameter effects of the optimal price and production run time.

The EOQ model assumes that the retailer's capitals are unrestricting and must be paid for the items as soon as the items are received. However, this may not be true. In practice, the supplier will offer the retailer a delay period, which is the trade credit period, in paying for the amount of purchasing cost. Before the end of the trade credit period, the retailer can sell the goods and accumulate revenue and earn interest. A higher interest is charged if the payment is not settled by the end of the trade credit period. Therefore, it makes economic sense for the retailer to delay the settlement of the replenishment account up to the last moment of the permissible period allowed by the supplier. In a real world, the supplier often makes use of this policy to promote his commodities. Huang et al. (2005) [4] investigated the optimal retailer's replenishment decisions under two levels of trade credit policy within the economic production quantity (EPQ) framework to reflect this realistic business situations. His Theorems help the retailer in accurately and quickly determining the optimal replenishment decisions under minimizing the annual total relevant cost. In this paper following inferences are made.

i. When replenishment rate is increasing, the optimal cycle time for the retailer will be decreasing. The retailer will order less quantity since the replenishment rate is faster enough.
ii. When the customer's trade credit period offered by retailer is increasing, the optimal cycle time for the retailer will be increasing. It implies that the retailer will order more quantity to get more interest earned offered by the supplier to compensate the loss of interest earned from longer trade credit period offered to his/her customer.
iii. Optimal cycle time for the retailer will be decreasing when the unit selling price s is increasing. This result implies that the retailer will order less quantity to take the benefits of the trade credit more frequently.

Freimer et al. (2006) [5] considered the classical economic production quantity model with defects produced according to some time-varying function u(x). If repair takes some constant time then the analysis either remains the same if units are considered to count in inventory when they are first produced, or the inventory expression is slightly modified if units are considered to count in inventory when the final, non-defective unit is produced.

The results hold for any random yield function as long as units are repaired instantly. With a deteriorating process, the optimal run length is shorter than the run length implied by the EPQ. Furthermore, the faster the process deteriorates the shorter the optimal run length. For the special case of linear deterioration, the cost penalty for using the EPQ instead of the optimal quantity is increasing in the setup cost, the defect repair cost and the rate of process deterioration, and decreasing in the holding cost. They consider the opportunity to invest in reducing setup cost and improving process quality. For a general time varying defect rate u(x), the marginal value of setup cost reduction is inversely proportional to optimal run length while the marginal value of process improvement (as modeled by a constant scaling p of the defect rate) is increasing in the optimal run length. Any investment in setup cost reduction will result in a reduction in the number of defects produced. Interestingly, the total number of defects can increase or decrease with the scaling p (probability of a defective unit). For the logarithmic investment function investigated, the number of defects at the optimal run length is unaffected by the holding cost and the setup cost. These results provide guidance for manufacturing managers regarding the relative value of setup cost reduction and process improvement investments.

This work allows any random yield function as long as all defective units are repaired instantly at some per unit cost. If all units are not repaired instantly, then one must allow for the possibility of lost sales or backorders, or impose conditions on the random yield function such that the effective production rate always exceeds the demand rate. It should be noted that while this approach generalizes several prior methods, some recent research has addressed other aspects of the EPQ problem such as stochastic demand, coordination between buyer and seller and choosing an inspection schedule. They focus on systems in which the process deteriorates over time, although they point out where our results may be generalized (in the opposite direction) for a process that improves over time. They develop properties of the optimal production quantity and optimal cost and consider the options of investing in reducing setup cost and improving process quality. Expressions for the marginal value of setup cost reduction and process improvement are also developed here.

Lai et al. (2006) [6] assumed that supplier would offer the retailer partially permissible delay in payments when the retailer ordered a sufficient quantity. Otherwise, permissible delay in payments would not be permitted. Under this condition, they developed the retailer's inventory system and develop three theorems to efficiently determine the optimal lot sizing decisions for the retailer. In this paper shortages were not allowed, time horizon was taken Infinite and replenishments were instantaneous.

In the paper Leung et al. (2007) [7] established more general results using the arithmetic-geometric mean inequality in which a general power function is proposed to model the relationship between production set-up cost (which implicitly measures the degree of process flexibility) and process reliability as independent variables and interest and depreciation cost as a dependent variable. Their objective was to minimize the long-run expected average annual cost function, i.e. the sum of setup, production, inventory holding, and interest and depreciation costs, which was a function of setup cost, production quantity, process reliability and the relevant cost parameters in the present case.

Due to the general nature of the power function proposed here, closed-form optimal solutions to this particular type of EPQ problem were not easy to obtain using the calculus-based optimization technique. To solve the EPQ problem presented in this paper GP approach was used where a closed-form optimal solution was obtained from GP. This was due to the fact that the objective function of the EPQ problem to be minimized was consisted of polynomials.

In this paper it was assumed that product quality is not always perfect; it is directly affected by the capability of the production process employed to manufacture a product and the quality assurance program established to monitor the product quality. Here, instead of measuring quality as the probability of the process going out of control with the production of the next unit, quality was measured as the expected fraction of a lot that is acceptable.

They took the assumption that 100% inspection is performed for each lot and no rework of defective items is possible. This assumption is valid for a process producing finished products, which are normally subject to 100% inspection and are irreversible.

Another assumption was that the total cost of interest and depreciation per production cycle is inversely related to the set-up cost and directly related to process reliability according to the following general power function. This assumption is based on the fact that to reduce the costs of production set-up and scrap and rework on shoddy products, substantial investment is required in improving the flexibility and reliability of the production process. Consequently the total cost of interest and depreciation per production cycle of the flexible production process should be much higher than that of conventional inflexible processes. However, though this relationship should be discrete but a continuous function was used as an approximation which was needed to simplify the subsequent mathematical analysis.

Liao et al. (2007) [8] derived a production model for the lot-size inventory system with finite production rate, taking into consideration the effect of decay and the condition of permissible delay in payments. Here, restrictive assumption of permissible delay is relaxed to that at the end of the credit period. The retailer will make a partial payment on total purchasing cost to the supplier and pay off the remaining balance by loan from the bank. At first, this paper shows that there exists a unique optimal cycle time to minimize the total variable cost per unit time. Then, a theorem is developed to determine the optimal ordering policies and bounds for the optimal cycle time are provided to develop an algorithm. Numerical examples reveal that our optimization procedure is very accurate and rapid. Finally, it is shown that the model developed by Huang [4] can be treated as a special case of this paper.

Here, demand rate and replenishment rate are considered known and constant. No shortages were allowed. The constant fraction of on hand inventory gets deteriorated per unit time and time period is taken infinite. The results in this study provide a valuable reference for decision-makers in planning the production and controlling the inventory.

Moreover, it provides a useful model for many organizations that use the decision rule to improve their total operation cost in real world.

Islam et al. (2007) [9] developed an EPQ model with flexibility and reliability consideration of production process and demand dependent unit production cost. The model has involved one storage space constraint. In real life problems, it is almost impossible to predict the restricted resource amount precisely. Decision maker may change it within some limits as per the demand of the situation. Hence it may be assumed uncertain in non-stochastic sense but fuzzy in nature. In this situation, the inventory problem along with constraint can be developed with fuzzy entries. The model is formulated in fuzzy environment introducing fuzziness in objective and constraint goals, coefficient and indexes of objective function and constraint. Shortages are not permitted and time horizon is infinite in this model. The unit production cost is considered as a continuous function of demand. Total cost of interest and depreciation per production cycle is inversely related to a setup cost and directly related to production process reliability. The total average cost of the inventory system consists of the set- up, production, inventory carrying and interest and depreciation cost. The problem is proposed to solve by Fuzzy Geometric Programming (FGP) method. The FGP method provides an alternative approach to this problem. Its advantages lie in its computational efficiency and in the primal–dual relationship. The method is efficient and reliable. Here decision maker may obtain the optimal results according to his expectation. The method presented is quite general and can be applied to the model in other areas like structural optimization, etc.

In the paper of Darwish et al. (2008) [10], the classical EPQ model is generalized by considering a relationship between the setup cost and the production run length. Previously it was considered that setup cost is fixed. However, setup time/cost is usually a function of the production run length. Processes with short production runs require less setup time/cost than that of long runs. This is because the effort needed to perform a setup activity is related to the condition of the production process. That is, for long production runs, the production process is more likely to be subjected to higher level of deterioration resulting in a higher setup time/cost. Thus, the setup cost and run length are correlated. In this paper, the classical EPQ model is generalized by considering a relationship between setup cost and production run length.

Deteriorating processes can be found in many applications, for example, plastic industry, and food processing and machine industry. The relationship between setup cost and production run length is also influenced by the learning and forgetting effects. Learning in setup encourages smaller lots (and consequently shorter runs) to be produced more frequently. The effect of forget ting in setup, however, is expected to have an opposite impact on the production run length because long production runs increase forgetting, which results in higher setup time/cost. Hence, the dependency between the setup cost and run length is related to process deterioration and learning and forgetting effects. Two models are developed, in one case shortages are not allowed and the other one permits shortages. Model 1 proposed EPQ model without backorders. It assumes that all demands are satisfied from inventory, so no stock out situation occurs. Model 2 proposed EPQ model with backorders. The model 1 presented here never becomes out of stock when a demand is occurred. In model 2, it is allowed for the system to be out of stock when a demand occurs is demonstrated. In such a case the demands occurring when the system is out of stock are backordered.

The model considered a process producing a single item at a rate P to satisfy a constant and deterministic demand. For each production cycle of length, the production process undergoes a setup at a cost. Other system parameters are assumed to be known with certainty and are independent of produced quantity. It is also assumed that the planning horizon is infinite and shortages are backlogged.

The cost functions associated with these models are proved to be convex and optimal solutions are determined. The results show that the setup shape parameter e, which is a property of the system, defines the category of the production system under consideration. For example, when e is more than unity, the optimal solution gives the least possible lot size, which is one unit. However, for a setup shape parameter approaching unity from below, the system falls into the category of producing in smaller lots more frequently. Moreover, values of e close to zero correspond to systems with relatively larger lot size and longer production run. Numerical results indicated that the loss due to using the classical EPQ model is significant. The results also show that the lot size and production run length are inflated when the relationship between setup cost and production run length is ignored. Thus, this model's credit is that it considered the setup cost as a function of the production run length which none of the previous EPQ models did.

Rau et al. (2008) [11] presented a new integrated production–inventory policy under a finite planning horizon and a linear trend in demand. It is assumed that the vendor makes a single product and supplies it to a buyer with a non-periodic and just-in-time (JIT) replenishment policy in a supply chain environment. The objective is to minimize the joint total costs incurred by the vendor and the buyer. In this study, a mathematical model is developed and proved that it has the optimal solution. Later, an explicit solution procedure for obtaining the optimal solution is described. Finally, two numerical examples is provided to illustrate both increasing and decreasing demands in the proposed model, and showed that the performance of the integrated consideration is better than the performance of any independent decision from either the buyer or the vendor.

In a competitive industrial environment, the integration to obtain an optimal production or inventory policy in the supply chain has become essential. This paper presents an integrated production inventory policy for a linear trend in demand with a non periodic replenishment policy in a supply chain under a finite time horizon. Most of the previous study considered the situation of constant demand; few have studied other demand patterns. But, the demand rate with a time-varying pattern reflects the actual environment. This study deals with a production– inventory policy integrating between a buyer and a vendor under a supply chain environment, and the contribution of this paper is that it proves the solution of an integrated model is the optimal solution.

Thus, in the proposed model, a single-buyer and a single vendor with a single item are considered and the demand is taken as a linear function. The production rate is constant for the vendor. It is greater than the demand at the end o f planning horizon for the case with increasing demand and less than the demand at the beginning of planning horizon for the case with decreasing demand. Due to preparing the first replenishment, it is considered that the vendor's planning horizon starts with a production lead time ahead of the buyer's planning horizon when the first delivery occurs. Delivery time considered negligible, shortages are not allowed and no stock was held at the beginning and the end of the time horizon.

In results, it is found that the joint total cost for the vendor and the buyer is a convex function. They proved that the model has the optimal and unique solution for both increasing and decreasing demands with several precious properties. In addition to illustrating the proposed model, the provided numerical examples suggest that the integrated approach is an effective way for partners to cooperate in a supply chain.

Panda et al. (2008) [12] developed a mathematical model for a single period multi-product manufacturing system of stochastically imperfect items with continuous stochastic demand under budget and shortage constraints. Here, the inventory system is an imperfect production system and involves multiple items. Their model is a single period inventory model where production rate is considered finite and constant and total demand over the period of cycle is considered stochastic and uniform over time. Shortages were permitted and fully backlogged and screening costs for all items considered same. After calculating expected profit in general form in terms of density functions of the demand and percentage of imperfectness, particular expressions for those density functions are considered. Here the constraints are of three types: (a) both are stochastic, (b) one stochastic and other one imprecise and (c) both imprecise. The stochastic constraints have been represented by chance constraints and fuzzy constraints in the form of possibility/necessity constraints. Stochastic and fuzzy constraints are transformed to equivalent deterministic ones using 'here and now' approach and fuzzy relations respectively. The deterministic problems are solved using a non-linear optimization technique-Generalized Reduced Gradient Method. The model is illustrated through numerical examples. Sensitivity analyses on profit functions due to different permitted aspiration' and 'confidence' levels are presented.

The paper proposed an extension to economic production lot size model for imperfect items in which the production rate is assumed to be finite and demand rate is stochastic under uncertain budget and shortage constraints. Here it is considered that the percentage of defective items is stochastic and the natures of uncertainty in the constraints are stochastic and/or fuzzy. Later, the stochastic non-linear programming problem was converted into a deterministic problem. It is considered that the density function of the demand is linear, which is a general case of uniform distribution. The advantage of the model is that it can be formulated and solved considering different types of density function.

Hejazi et al. (2008) [13] determine the economic production quantity with reduced pricing, rework and reject situations in a single-stage system in which rework takes place in each cycle after processing to minimize total system costs. The assumption considered in this paper is that processing leads to different products classified in the four groups of perfect products, imperfect products, defective but reworkable products, and finally non-reworkable defective products. The percentage of each type is assumed to be constant and deterministic. A mathematical model is developed and numerical examples are presented to illustrate the usefulness of this model.

This paper extends the work by Jamal et al. [14] and studies the optimal run time problem of EPQ model with imperfect products, reworking of the reparable defective products and rejecting of non-reworkable defective items. Neglecting the production of imperfect products and scraps, Jamal et al. assumed that all defective products could be reworked, but in some real situations, it is observed that some non-perfect products cannot be reworked and they should be either sold at a lower price or rejected altogether. In this paper, the different scenarios for imperfect quality products are investigated. While previous assumption was that defective items are reworked instantaneously with processing at no additional time, their assumption is that these products should be reworked at the end of the processing period in a kind of reprocessing stage. In other words, reworking an imperfect item takes time and money as does the processing of a product. In the paper, it is considered that the lot contains a percentage of defectives, so that these defective products can be reprocessed, or reworked, after the processing period and kept in stock. These products are assumed to be of good quality after reprocessing. Thus, the reworked products will need no inspection. Each lot produced also contains a percentage of defectives, so that these units are rejected with an associated cost when identified. In other words, a defective product that cannot be reworked is rejected immediately after its work operation completes with an associated cost. Finally, the objective is to minimize the total system cost of the inventory system.

Chen et al. (2008) [15] introduced a Fuzzy Economic Production Quantity (FEPQ) model with defective productions that cannot be repaired. This FEPQ model is applicable when inventory continuously flows or builds up over a period of time after an order has been placed, when units are produced and sold simultaneously. In this model, a fuzzy opportunity cost and trapezoidal fuzzy costs under crisp production quantity or fuzzy production quantity is considered in order to extend the traditional production inventory model to the fuzzy environment. The authors use Function Principle as arithmetical operations of Fuzzy Total Production Inventory Cost (FTPIC), and use the Graded Mean Integration Representation method to defuzzify the fuzzy total production and inventory cost. Then they use the Kuhn–Tucker method to find the optimal economic production quantity of the fuzzy production inventory model.

Throughout this paper, the authors only use normal trapezoidal fuzzy number as the type of all fuzzy parameters in our proposed fuzzy production inventory models. In real world, defective products cannot be avoided in some production processes. So, the model considered defective products. Two cases of imperfect productions that cannot be repaired are considered, one case with fuzzy costs but crisp production quantities, the other case with fuzzy costs and fuzzy production quantities.

Chung et al. (2009) [16] proposed an inventory model which has threefold purpose:

i. This paper shows that the total cost function per unit time is convex by a rigorous proof.

ii. This paper derives the closed forms for the upper and lower bounds on the optimal cycle time of the total cost function per unit time, thereby enabling straightforward application of the standard bisection algorithm to numerically compute the optimal cycle time.
iii. This paper compares optimal solutions obtained by using the bisection algorithm and Park's approach. Numerical examples show that bisection algorithm approach is better.

For each raw material, here it is considered that a constant fraction of the on-hand inventory decays per unit time and there is no replacement of the decayed inventory. The decay of the raw materials is assumed to be a constant fraction of the on-hand inventory. The product is produced in batches and the raw materials were obtained from outside suppliers. The objective is to minimize the total cost of the system and to study the convexity (concavity) of the total annual cost (profit) function of the inventory model to locate the optimal solution.

In 2007**,** Huang et al.[17] proposed the optimal retailer's replenishment decisions in the EPQ model under two levels of trade credit policy, in which the supplier offers the retailer a permissible delay period M , and the retailer in turn provides its customer a permissible delay period N (with N < M). Teng and Cheng (2009) [18] extend his EPQ model to complement the shortcoming of the model. In addition, they relax the dispensable assumptions of N < M and others. They considered two cases. In one case, the manufacturer buys all parts at time zero and must pay the purchasing cost at time M. Based on the time at which the manufacturer must pay the supplier to avoid interest charge (M) and the time at which the manufacturer receives the payment from the last customer (N), two possible sub-cases are created. In Sub- case 1, the manufacturer pays off all units sold by M-N at time M, keep the profits and starts paying for the interest charges on the items sold after M - N. In this sub-case 2, the manufacturer receives the total revenue, and is able to pay the supplier the total purchase cost at time M. In case 2, the customer's trade credit period N is equal to or larger than the supplier credit period M. Consequently, there is no interest earned for the manufacturer. In addition, the manufacturer must finance all items ordered at time M at an interest charged per dollar per year, and start to pay off the loan after time N.

Both Huang and Cheng models are to investigate the optimal replenishment policies in the EPQ model under two levels of trade credit financing. Both models assume that the manufacturer buys and receives all parts at time zero and must pay the purchasing cost at time M, which is the time the manufacturer must pay the supplier in full to avoid interest charge. Since the manufacturer offers its customers the permissible delay of N periods, the manufacturer starts receiving its revenue at time N. Huang' s model did not recognize that the last customer buys the product at time T, and pays the manufacturer at time T + N due to its customer trade credit period N. In addition, Huang's model ignored the fact that the manufacturer starts getting the revenue at time N, not at time 0.

Maiti et al. (2009) [19] introduced an EPQ model for a deteriorating item with linearly displayed demand in imprecise environment (involving both fuzzy and random parameters) under inflation and time value of money. Here, it is assumed that the periods of business are random and follow exponential distribution with a known mean. So, the resultant effect of inflation and time value of money is assumed as fuzzy in nature. A particular case is also analyzed where resultant effect of both inflation and time value is crisp in nature. For crisp inflation effect, the total expected profit for the planning horizon is maximized using the Genetic Algorithm (GA) to derive optimal inventory decision. On the other hand when inflationary effect is fuzzy then the expected profit is fuzzy in nature too. For crisp model expected profit is proposed to maximize using a GA with roulette wheel selection, arithmetic crossover and random mutation. In the case of fuzzy model, a fuzzy simulation process is proposed to maximize the optimistic/pessimistic return of the objective function and a fuzzy simulation based genetic algorithm with GA operators is developed to solve the model. Model-1 is solved using GA. and model-2 is solved by converting the possibility/necessity constraint to its deterministic equivalent.

Islam et al. (2009) [20] incorporates inflation and time value of money in EPQ model for a newly launched product. In the paper, the demand of the item is displayed stock dependent and lifetime of the product is random in nature and follows exponential distribution with a known mean. Here learning effect on production and setup cost is incorporated. Model is formulated to maximize the expected profit from the whole planning horizon. A fuzzy-based lifetime extension of genetic algorithms is considered. A genetic algorithm with varying population size is used to solve the model where crossover probability is a function of parent's age type (young, middle-aged, old, etc.). In this GA a subset of better children is included with the parent population for next generation and size of this subset is a percentage of the size of its parent set. This GA is named fuzzy genetic algorithm (FGA) and is used to make decision for above production inventory model in different cases. The model is illustrated with some numerical data. Sensitivity analysis on expected profit function is also presented. Performance of this GA with respect to some other GAs is compared. An EPQ model has been considered under inflation and time discounting over a stochastic time horizon incorporating the learning effect on both the production and setup cost.

Panda et al. (2009) [21] developed multi item EPQ models with price dependent demand, infinite production rate, stock dependent unit production and holding costs. Here, flexibility and reliability consideration are considered also. The models are developed under two fuzzy environments. First one is with fuzzy goal and fuzzy bindings on storage area and the other one is with unit cost as fuzzy and possibility of necessity restrictions on storage space. The objective goal and constraint goal are defined by membership functions. The presence of fuzzy parameters in the objective function is dealt with fuzzy possibility/necessity measures. The first one-the fuzzy goal programming problem is solved using fuzzy additive goal programming (FAGP) and modified geometric programming (MGP) methods. The second model with fuzzy possibility/necessity measures is solved by geometric programming (GP) method. The models are formed as maximization problems. In the paper, they have formulated multi-item profit maximization production inventory models with limited storage area in fuzzy/fuzzy possibility and necessity sense under process reliability and flexibility. They have considered fuzzy possibility and necessity measures of the objective function when its some parameters are fuzzy. Moreover, the authors have considered demand as power function of selling price, unit costs dependent on inventory level, holding costs again as functions of unit costs.

Hu et al. (2010) [22] investigated the optimal replenishment policy under conditions of permissible delay in payments and allowable shortages within the EPQ framework. Considering a practical situation, the unit selling price is not lower than the unit purchasing price and the infinite replenishment rate is difficult to reach in general, the authors extend the work of Chung and Huang [15] to assume that the replenishment rate is finite and the unit selling price is not necessarily equal to the unit purchasing price. They assume that replenishment rate is finite and the unit selling price is not necessarily equal to the unit purchasing price

The necessary assumptions made here are: (1) Demand rate and replenishment rate are both known and constant. (2) Time horizon is infinite. (3) The supplier proposes a certain credit period M. During the time the account is not settled, the retailer deposits his/her. At the end of the trade credit period, the account is settled and the retailer starts paying for the interest charges on the items in stock (including negative stock and positive stock). (4) Shortages are allowed and are fully backlogged. In the paper, the inventory cycle is divided into four major phases: backorder replenishment period, inventory building period, inventory depletion period and shortage period. The objective here is to find the optimal replenishment time and the corresponding backorder replenishment period, which minimize the annual total relevant cost.

In this paper some realistic features are considered. First, the unit selling price and the unit purchasing price are not necessarily equal to match the practical situations. Second, the replenishment rate is finite to make a broader application scope. Third, since stock out is unavoidable due to various uncertainties in many practical situations, it is assume that the shortages are allowed. A theorem is developed to determine the optimal replenishment policy. Finally, numerical examples are given to illustrate the theorem.

Leung et al. (2010) [23] generalized a number of integrated models with/without lot streaming and with/without complete backorders under the integer–multiplier coordination mechanism. They then individually derived the optimal solution to the three- and four-stage model, using algebraic methods of complete squares and perfect squares. It is subsequently deduced optimal expressions for some well-known models. For this model, Leung found that the optimal solution is a global one, which is algebraically derived.

Two extensions of the model augmenting future research endeavors in this field are: First, following the evolution of three- and four-stage multi-firm supply chains, can the integrated model of a five- or higher-stage multi-firm supply chain is formulated and algebraically analyzed. Secondly, using complete and perfect squares, we can solve the integrated model of a stage multi-firm supply chain either for an equal cycle time, or an integer multiplier at each stage with a fixed ratio partial backordering allowed for some/all downstream firms, with or without lot streaming can be solved.

Saadany et al. (2010) [24] developed and analyzed production, remanufacture, and waste disposal EPQ type models, where a manufacturer serves a stationary demand by producing new items of a product as well as by remanufacturing collected used/returned items. In these developed models, the return rate of used items is modeled as a demand-like function of purchasing price and acceptance quality level of returns. The model developed herein is a decision tool that helps managers in determining the optimum acceptable acquisition quality level and its corresponding price for used items that are collected for recovery purposes and that minimizes the total system cost.

Two mathematical models were developed. The first assumes a single remanufacturing cycle and a single production cycle, with the second being a generalized version of the first assuming multiple remanufacturing and production cycles. A solution procedure was introduced with an enhanced search technique that eliminates solution branches that do guarantee an optimal solution. This enhanced solution procedure was supported by a theorem, which shows that having even numbers of remanufacturing (m) and production (n) cycles in an interval never produces an optimal solution. Numerical results showed that when considering the return rate of used items to be dependent on the purchasing price and acceptance quality level of these returns, a pure policy of either no waste disposal (total repair) or no repair (total waste disposal) is not optimal. Results showed that a mixed (production + remanufacturing) strategy is optimal, when compared to either a pure strategy recycling (pure remanufacturing) or a pure strategy production.

An immediate extension of the work presented herein is to integrate the production–remanufacturing system into a multistage supply chain (say supplier–manufacturer–retailer), where used items are collected from the market by the manufacturer to be disassembled for reuse. In this case, the production–remanufacturing process will be supplied by components from the disassembly process and from the manufacturer's supplier as needed. A second extension is to assume that the production and remanufacturing processes are imperfect where defective items are either reworked or scrapped. A third extension is to assume demand to be stochastic.

Sana et al. (2010) [25] considered a production–inventory model in an imperfect production process over a finite planning horizon. The production rate is a dynamic variable (i.e., varying with time). Increasing the time-varying demand (like, quadratic, linear, exponential and stock-dependent) is considered. And, the unit production cost is considered as a function of production rate and product reliability parameter. The integrated profit function with the effect of inflation and time value of money is maximized by Euler–Lagrange's method.

The model of Sana provides a guide for a firm/industry in addressing the question- when and in what to invest to maintain sustainable competitive advantage? The firm/industry produces a single product and operates in an oligopolistic competition. Demand for the product in an industry depends on price, time and performance quality with time. The effect of this dependency is that the retailers have incentive to keep higher levels of inventory in spite of higher holding costs as long as the item is profitable and the demand is an increasing function of the inventory-level. Increasing production knowledge decreases unit production cost whereas lower values of product reliability factor increases development cost. Therefore, productivity and quality knowledge can be developed through induced and autonomous learning in order to strengthen company position.

In the paper of Ata Allah et al. (2010) [26] an EPQ model with multiple discrete deliveries, capacity and space constraints is presented which solved by using the extended cutting plane method, the particle swarm optimization (PSO) and harmony search algorithms. Here the research objective is to determine the optimal period length, the optimal number of shipments and the optimal order quantities to minimize the total production inventory cost with space and capacity constraints. It is assumed that production and demand rates of each product are known and constant. Manufacturer sends orders to the customer and bears the transportation cost for each delivery to the customer. The customer determines the capacity of each delivery and the quantity of each shipment. Shortage is not permitted and the production costs consist of production, setup, holding, and transportation costs. Since all products are manufactured by a single machine with a limited capacity, a unique cycle length for all items is considered. The final model is a mixed integer nonlinear programming (MINLP) problem and extended cutting plane method to solve MINLP is used. In addition, in order to evaluate the performance of the proposed solution method, two meta-heuristic algorithms are used. Two numerical examples with fifteen products are used to illustrate the proposed model. Through the numerical examples, it is demonstrated that the extended cutting plane method performs better in terms of the objective function and the computation time. The examples also show that high holding cost and production cost result in less number of shipments in each cycle.

The proposed PSO algorithm of this paper consists of three main steps: firstly, the positions of particle are generated. Secondly, exploration velocity is updated, and finally each position is updated. Here, each particle refers to a point in the solution space that changes its position from one move (iteration) to another, based on exploration velocity updates. The type of particles is associated with the number of variables involved in problem.

Wang et al. (2009) [27] integrates fuzzy simulation and PSO algorithm to solve an EPQ model where the fuzzy simulation is employed to estimate the –level minimal average cost, and PSO algorithm is used to find the optimal solution. He consider the EPQ problem with backorder in the fuzzy sense, where the setup cost, the holding cost and the backorder cost are characterized as fuzzy variables respectively. As general extensions of the classical EPQ model, a fuzzy EVM and a fuzzy CCP model are constructed, respectively. The EVM can be solved with generic approaches; however the CCP model needs to be solved with the heuristic algorithm owing to the complexity of the problem. There are several heuristic algorithms inspired from the evolution of nature, such as Evolutionary Computation (EC) technique, GA and PSO algorithm. To solve the CCP model, PSO algorithm is selected as the foundation to design an algorithm which integrates fuzzy simulation and PSO algorithm. In this paper, only production cost, holding cost, backorder cost and setup cost are considered.

In EPQ model when objective function becomes complex and sophisticated in nature like nonlinear non-integer-programming model, reaching an analytical solution (if any) is difficult and time consuming. As a result, meta-heuristic search algorithms were successfully used by many researchers to solve such type model. Many researchers have successfully used meta-heuristic methods to solve complicated optimization problems in different fields of scientific and engineering disciplines. Some of these meta-heuristic algorithms are simulating annealing, threshold accepting, Tabu search, GA, neural networks, ant colony optimization, fuzzy simulation, evolutionary algorithm and harmony search. Seyed et al. (2009) [28] developed a multi-product EPQ model in which there are some imperfect items of different product types being produced such that reworks are allowed and that there is a warehouses pace limitation. Under these conditions, they formulate the problem as a nonlinear integer-programming model and propose a genetic algorithm to solve it. Three main specifications of the proposed model of this research that have led to its novelty are (1) the allowance of several products, (2) rework and imperfect product are allowed, and (3) the warehouse space to store raw materials and finished goods is limited. By allowing these conditions simultaneously, their model has demonstrated its difference from the other models in the EPQ literature. At the end, a numerical example is also presented to identify the optimal values of the genetic algorithm parameters and to illustrate the applications of the proposed methodology to more realistic real- world problems.

One major assumption in previous studies is that item quality is perfect. However, in reality, product quality is not always perfect, and often depends on the reliability of the production process used to produce the product. An important managerial decision therefore concerns investment to improve process reliability and quality. Investment decisions to improve process reliability and quality have been investigated by several researchers in the cost-minimization context. Tripathy et al. (2011) [29] consider this reliability/quality issue in the context of linking production (lot sizing and inventory) and marketing (pricing) decisions. Authors address some modeling issues of a related previous study. They develop a profit-maximization model, which is investigated under two different decision-making approaches to linking production and marketing (sequential and joint decisions). Calculus and geometric programming are used to solve the model, to compare the decisions from the two approaches analytically, and to develop a practical heuristic approach. The comparative results show that decision patterns can be the opposite of the optimal decisions for the perfect-quality case reported in the literature.

The main purposes of this study are twofold. First, Authors address issues related to the model of Cheng (1991) [30] and improve the model. Secondly they investigate the economic relationship between price, demand rate, lot size, and reliability level in a broader decision-making context that links production and marketing functions. The objective is profit maximization rather than cost minimization. This is because optimization is trivial when production costs are minimized and demand rate is a decision variable. The authors develop profit-maximization models that treat the product price as a decision variable and the demand rate as an intervening variable to be determined by the price.

Sadjadi et al. (2010) [31] proposes a new method where the reliability of the production is incorporated into pricing, marketing and production planning. The integrated model of this paper simultaneously determines price of products, marketing expenditure, lot size, setup cost, inventory holding cost and reliability of the production process. The objective is to minimize total costs including marketing, production, setup, holding, and interest and depreciation costs. This model is formulated in GP form and the optimal solution in closed form is determined using the art of GP technique. This model is formulated as a nonlinear optimization problem and the optimal solution in closed form is derived using geometric programming. In order to examine the behavior of the proposed method, the study tests the modeling formulation using a numerical example.

The process reliability depends on a great variety of factors such as production technology, machine capability, jigs and fixtures, work methods, use of on-line monitoring devices, skill level of the operating personnel and inspection, maintenance and replacement policies. Higher reliability means products with acceptable quality are more consistently produced by the process; thereby reducing the costs of scrap and rework of substandard products, wasted materials and labor hours. However, high reliability can be achieved with substantial capital investment, which increases the cost of interest and of the depreciation of the production process. The total cost of interest and depreciation per production cycle is inversely related to the set-up cost and directly related to process reliability according to the following general power function. This assumption is based on the fact that to reduce the costs of production set-up and scrap and rework on shoddy products, substantial investment is required in improving the flexibility and reliability of the production process. In reality, this relationship should be discrete but a continuous function is used as an approximation which is needed to simplify the subsequent mathematical analysis.

So far depreciation and insurance cost was considered with production cost, setup cost and holding cost. Some paper considers cost of quality such as inspection and rejection cost along with basic inventory costs (setup cost, holding cost and production cost). A very few paper considers backorder cost with traditional EPQ model. Some paper considers process reliability but only for basic inventory cost terms. Thus, these papers do not consider the combined effect of production cost, setup cost, holding cost, inspection cost, depreciation cost, rejection cost and backorder cost on total cost minimization. This thesis work intends to consider all these costs with reliability of the production process. A meta-heuristic PSO algorithm will be used to solve the unconstrained non integer non linear programming model as it can generate accurate result with a shorter computational time. Finally, the results obtained from PSO algorithm will be compared with GA applying on the same inventory model.

CHAPTER 3
MODEL FORMULATION

3.1 Problem Identification

A basic assumption in the inventory management system is that set-up cost for production is fixed. In addition, the models also implicitly assume that items produced are of perfect quality. However, in reality, products are not always perfect but are directly affected by the reliability of the production process employed to manufacture the product. The process reliability depends on a great variety of factors such as production technology, machine capability, jigs and fixtures, work methods, use of on-line monitoring devices, skill level of the operating personnel and inspection, maintenance and replacement policies. Process reliability thus related with production cost, holding cost, inspection cost, depreciation and interest cost, rejection cost and even with backorder cost. Backorder cost plays an important part in increasing the cost of production which was ignored so far or only linear backorder cost was considered. Actually, traditional inventory model with cost of quality (inspection and rejection cost), backorder cost, setup cost, depreciation cost with context of reliability is a most realistic phenomena for a production process.

3.2 Problem Definition

The purpose of this research is to extend the previous research in the EPQ model by employing the knowledge of cost of quality, backorder cost and reliability of production process. In this thesis work, demand rate, production capacity and inventory holding cost of the product are known parameter and the production process is assumed to be not 100% perfect, i.e. a fraction of the produced items are defective. It is assumed that the defective items are sold at a reduced price and the selling price of fresh units is taken as a mark-up over the unit production cost. The model is formulated to determine the optimal reliability, lot size and cycle time in order to minimize the total cost of production**.** Mathematical equations are obtained and derived for production cost, holding cost, setup cost, inspection cost, depreciation cost, interest cost, rejection cost, backorder cost, etc. with reliability of the production process which is very important in real life production inventory problem**.** This paper also incorporated both administrative backorder cost and backorder cost due to goodwill loss. As a result, the inventory model in this paper is more practical than the traditional EPQ model. A new meta- heuristic particle swarm optimization (PSO) algorithm is used to solve the unconstrained non integer non linear programming model as it can generate more accurate result with a shorter computational time than any other meta-heuristic algorithms.

3.3 Assumption of the Study

Some assumptions are considered in this thesis works. Assumptions are as follows:

i. Preparation time is negligible.
ii. Production starts immediately after receiving the order.
iii. The demand for the imperfect product with reduced price always exits.
iv. The demand rate for the good product is deterministic and constant.
v. Production and demand rates (with the former greater than the latter) are independent of production or order quantity, and are constant.
vi. The replenishment lead time is zero.
vii. Backorders and lost sales are allowed

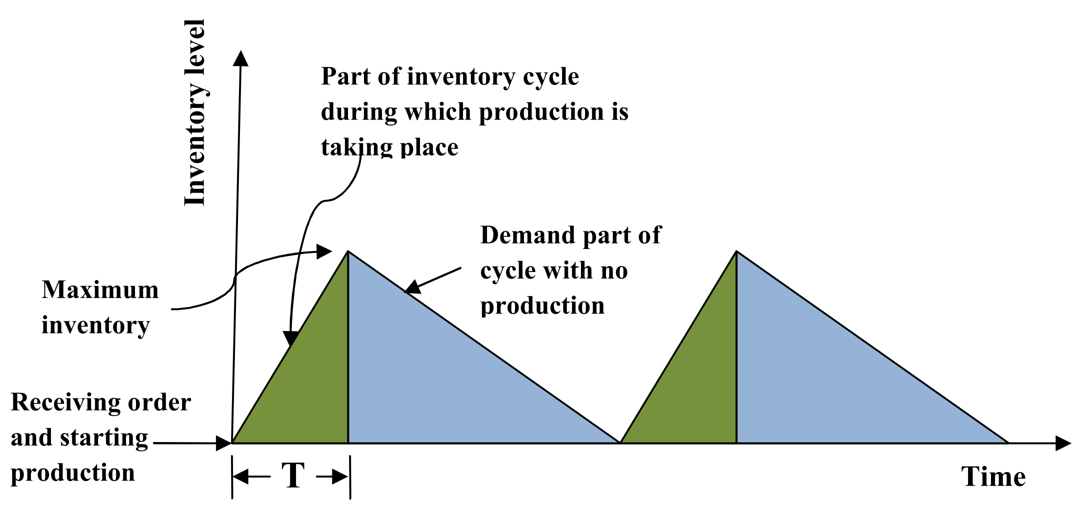

Figure 3.1: Instantaneous production in EPQ model

viii. Here, cycle time (T), production process reliability (r), lot size (Q) are decision variables.

ix. Defective items are sold immediately with a lower price than fresh items.

x. The production rate is greater than the demand rate, i.e. $>$.

Where *P* is production rate per day and *D* is demand per day which in deterministic in nature.

xi. Total cost of interest and depreciation per production cycle (,) is inversely related to set-up cost *S* and directly related to process reliability (r) according to following general power function [29]:

$$(\ , \ ) = \qquad (1)$$

Where *l*, *m* and *n* are positive constants chosen to provide best fit of the estimated cost function

3.4 Mathematical Modeling

In this study, a new model is being proposed which incorporates the reliability as part of an integrated model. The model simultaneously determines lot size (Q), production reliability (R), and cycle time (T). The total cost incurred in a production cycle is sum of set-up, production, inventory holding, rejection, inspection, backorder and interest and depreciation costs. The objective function is to minimize the total cost of per production cycle as follows:

Minimize Total cost = (Production cost) + (Setup cost) + (holding cost) + (Inspection cost) + (Depreciation and Interest cost) + (Cost of defective items) + (Administrative backorder cost) + (Backorder cost due to loss of goodwill) (2)

As with the classical EPQ model, we assume that the production cycle is repeated indefinitely with an infinite planning horizon and so can base our analysis on one typical production cycle. A level of process reliability *r* means that, of all the items produced in a production cycle, only *r* % are of a quality acceptable enough to meet demand. It means that – products must be produced to satisfy the whole demand. It is thus evident that the length of a production cycle is — and the number of cycles per year is—, where Q is the lot size in each cycle and D is the demand rate. Also, the average amount of inventory held each year is—. Therefore, the model can be formulated as follows:

3.4.1 Production cost

According to Tripathy, Wee and Majhi (2003), [32] the unit cost of production is directly related to reliability and inversely related to the demand where demand exceeds supply. Later, Tripathy and Pattnaik (2009) [33] studied that unit cost of production is inversely related to both process reliability and demand. Generally, total production cost is denoted as CD, where C is the unit production cost. But if reliability is considered, this equation will be changed. Due to increase of reliability production cost will increase because now manufacturer has to be produced – amount of products instead of regular demand D. So, it is clear that total production cost is inversely related with reliability of the production process.

Total production cost = — (3)

3.4.2 Setup Cost

Setup costs refer to expenses incurred each time a batch is produced. It consists of engineering cost of setting up the production runs or machines, paperwork cost of processing the work order, and ordering cost to provide raw materials for the batch. In most of the previous study, setup cost is considered as,

Setup cost = – (4)

Where S is the setup cost per cycle and T is the cycle time or production period in each cycle. In traditional EPQ and EOQ model cycle time T is represented as,

Cycle time, T =– (5)

Thus setup cost =— (6)

3.4.3 Holding Cost

Holding cost associated with keeping inventory, including warehousing, spoilage, obsolescence, interest and taxes; also called Inventory Carrying Costs. The inventory holding cost per cycle is obtained as the average inventory times holding cost per product per cycle. In traditional EPQ/EOQ model inventory carrying cost is considered as,

Holding cost =— (7)

In the study of Sakaguchi et al. (2009), [34] the holding cost is considered with reliability with a concept that holding cost will decrease with the increase of reliability. They derived the following equation:

Holding cost = —— [1 − —] (8)

Here, *P* is the production rate of the process.

3.4.4 Depreciation and Interest cost

To reduce the costs of production setup, scrap and rework on shoddy products, substantial investment is necessary in improving the flexibility and reliability of the production process. Consequently, the total cost of interest and depreciation per production cycle of the modern flexible production process is much higher than that of the conventional inflexible process. An increase in the reliability of the production thus process leads to growth in total interest and depreciation cost. This relationship can be realized easily, regarding the fact that high reliability can only be achieved with additional cost in practice. So, the total cost of interest and depreciation per production cycle is assumed as a function of reliability and setup cost [7]:

Depreciation and interest cost = (9)

and depreciation and interest cost per cycle = / (10)

Here *l, m* and *n* are positive constants chosen to provide best fit of the estimated cost function and *S* is the setup cost per cycle. Equation (10) also implies that when the setup cost is decreased the total costs of interest and depreciation also increase. This relationship is apparent and is based on the fact that manufacturers must invest heavily in order to reduce the setup cost per production. For instance, it may cost much more to decrease the unit setup cost, since the study needs to acquire expensive facilities.

3.4.5 Inspection Cost

Previous studies had considered that imperfect quality and defective items are either to be reworked instantaneously and kept in stock or rejected at a cost. In this study lower pricing, rework and reject situations are integrated into cost of defective items. A 100% inspection is performed in order to identify the amount of good quality items, imperfect or defective items in each lot. It is assumed that, inspection only takes place during the processing time. If *I* is the per unit inspection cost then,

Inspection cost = (11)

or, Inspection cost = — (12)

But if production process reliability is improved then less defective items will be produced and amount of product to be inspected will be reduced. With the increases of reliability, inspection cost will decrease. It can be further explained that with the decrease of reliability producer have to produce – items per year instead of .

So Inspection cost will be= — (13)

3.4.6 Cost of Defective Items

Jamal et al. [14] considered reworking of the reparable defective products and rejecting of non-reworkable defective items. Neglecting the production of imperfect products and scraps, Jamal et al. assumed that all defective products could be reworked, but in some real situations, it is observed that some non-perfect products cannot be reworked and they should be either sold at a lower price or rejected altogether [13]. In this paper, the defective items are sold in a reduced price. If the process reliability is r, then (1-r) % product is defective. If the cost of defective product is j then rejection cost per unit time will be

$$\text{Rejection Cost}=\frac{(\quad)}{} \qquad (14)$$

From the equation, it is clear that rejection cost decreases with the increase of production process reliability that is proportionally related. This comes from the fact that, a flexible and reliable production process produces less defective product.

3.4.7 Backorder Cost

When a customer seeks the product and finds the inventory empty, the demand can either go unfulfilled or be satisfied later when the product becomes available. The former case is called a lost sale, and the latter is called a backorder. Backorder cost includes freight cost, packaging material cost, warehouse processing cost, backorder notification cost, cancellation cost, lost demand, lost cost of initial customer acquisition, loss of future sells and goodwill, etc. We will consider all these backorder costs in two categories- administrative backorder cost and backorder cost due to loss of goodwill

3.4.7.1 Administrative backorder cost

Backorder administrative cost per unit reflects additional work needed to take care of the backorder. The number of backordered customers per year can be determined by recognizing that during each inventory cycle there are *Y* backorders and there are *D/Q* inventory cycles per year. Hence,

$$\text{Number of Backorders per Year} = \frac{}{} \qquad (15)$$

If backorder administrative cost per unit is , then

Administrative backorder cost= (Number of backorders during the year) × (Administrative backorder cost per unit)

$$\text{Administrative backorder cost} = \frac{}{} \qquad (16)$$

If reliability of the production process increases then there will be less chance of remaining unfulfilled customer demand and less backorder amount will come to the producer. So, it is assumed that reliability is inversely related with backorder administrative cost.

Backorder administrative cost considering reliability = —— (17)

3.4.7.2 Backorder cost due to loss of goodwill

Backorder cost per unit per year or loss of goodwill cost reflects future reduction in profitability. It can be estimated from market surveys and focus groups. The annual backorder cost is dependent on the average backorder level [35] and the number of backordered customers per year.

Average backorder level = — (18)

If annual backorder cost per unit is , then

Backorder cost due to loss of goodwill = —— (19)

Optimal backorder [35] is derived for general EOQ model as, *Y*= ——— (20)

Putting the value of *Y* in equation (17) and (19) and solving the equation,

Total backorder cost becomes = ——+ ——— + ——— - ——— – ——— (21)

Here, $z = h+\beta$. For convenience, we repeat the equation of total cost,

Total cost per cycle = (Production cost) + (Setup cost) + (holding cost) + **(**Inspection cost) + (Depreciation and Interest cost) + (Cost of defective items) + **(**Administrative backorder cost) + (Backorder cost due to loss of goodwill)

Combining the equations of different cost items we get, Total cost per cycle = —+ — + — [1 − —] + / + —+ ——( )——+ ——+ —— +

—— - —— – —— (22)

The objective is to minimize the total cost per cycle subject to the constraint that process reliability cannot exceed 100%. In the developed inventory model, too much constraints such as that defective items, back orders, reliability etc. has been considered. The model is in unconstrained non integer nonlinear in nature, reaching an analytical solution (if any) to the problem is difficult. In addition, efficient treatment of integer nonlinear optimization is one of the most difficult problems in practical optimization. Many researchers have successfully used meta-heuristic methods to solve complicated optimization problems in different fields of scientific and engineering disciplines; among them, the PSO algorithm is one of the most efficient methods. Here, the optimal solution is determined using the art of PSO technique. Later, the study analyzes the behavior of the model using some realistic example when different parameters are changed. Finally, the results obtained from PSO algorithm will be compared with GA applying on the same inventory model.

CHAPTER 4
INVENTORY MODELING WITH PARTICLE SWARM OPTIMIZATION (PSO)

In EPQ model when objective function becomes complex and sophisticated in nature like nonlinear non-integer-programming model, reaching an analytical solution becomes difficult and time consuming [10]. As a result, meta-heuristic search algorithms were successfully used by many researchers to solve such type model. Reja et al. (2009) [36] used genetic algorithm to solve multi-product economic production quantity model with defective items, rework, and constrained space. Maiti et al. (2008) [37] solved an EPQ model via Genetic Algorithm which deals with price discounted promotional demand in an imprecise planning horizon. A relatively new technique is particle swarm optimization (PSO) which has been successfully applied to a wide range of applications, but so far gets less attention as a meta-heuristic algorithm to solve such type of complex problem though it can generate high-quality solutions with shorter calculation time and stable convergence. As the model is unconstrained non integer non linear form which is complex in nature it requires a meta-heuristic search algorithm to optimize. In this thesis, PSO has been used to optimize the model. In the following section basic optimization procedures, core ideas of PSO and our developed PSO code will be discussed.

4.1 Optimization

The task of optimization is that of determining the values of a set of parameters so that some measure of optimality is satisfied, subject to certain constraints. This task is of great importance to many professions, for example, physicists, chemists and engineers are interested in design optimization when designing a chemical plant to maximize production, subject to certain constraints, e.g. cost and pollution. Scientists require optimization techniques when performing non- linear curve or model fitting. Economists and operation researchers have to consider the optimal allocation of resources in industrial and social settings. Some of these problems involve only linear models, resulting in linear optimization problems, for which an efficient technique known as linear programming exists. The other problems are known as non-linear optimization problems, which are generally very difficult to solve. This problem has been considered in this thesis. The term optimization refers to both minimization and maximization tasks. A task involving the maximization of the function f is equivalent to the task of minimizing $-f$, therefore the terms minimization, maximization and optimization are used interchangeably.

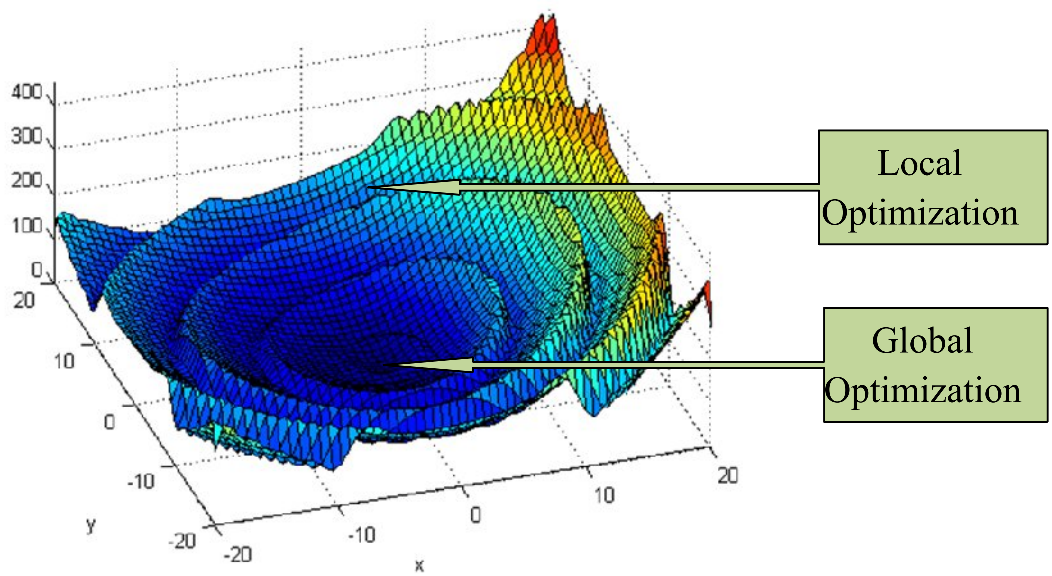

Figure 4.1: Local and Global optimization

Some problems require that some of the parameters satisfy certain constraints, e.g. all the parameters must be non-negative. These types of problems are known as constrained minimization tasks. They are typically harder to solve than their equivalent unconstrained versions, and are not dealt with explicitly here. The thesis deals with a unconstrained minimization tasks. Techniques used to solve the minimization problems defined above can be placed into two categories: Local and Global optimization algorithms. The figure (4.1) illustrates the concept of local optimization and global optimization. The deterministic local optimization algorithms include simple Newton-Raphs on algorithms, through Steepest Descent [38] and its many variants, including the Scaled Conjugate Gradient algorithm and the quasi- Newton family of algorithms. Some of the better known algorithms include Fletcher Reeves (FR), Polar-Ribiere (PR), Davidon-Fletcher-Powell (DFP) and Broyden-Fletcher-Goldfarb-Shanno (BFGS) [38].

4.2 Particle Swarm Optimization (PSO)

Let we consider two swarms flying in the sky, trying to reach the particular destination. The swarms generally based on their individual experience choose the proper path to reach the particular destination. Apart from their individual decisions, they take decisions about the optimal path based on their neighbor's decision and hence they are able to reach their destination faster. The mathematical model for the above mentioned behavior of the swarm is being used in the optimization technique named as the Particle Swarm Optimization Algorithm (PSO).

PSO is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. PSO optimizes a problem by having a population of candidate solutions and moving these particles around in the search-space according to simple mathematical formulae over the particle's position and velocity. Each particle's movement is influenced by its local best known position and it's also guided toward the best known positions in the search-space, which are updated as better positions are found by other particles. This is expected to move the swarm toward the best solutions.

Particle Swarm Optimization is a population- based optimization method which was first proposed by Kennedy and Eberhart [39]. Some attractive features of the PSO include the ease of implementation and the fact that no gradient information is required. It can be used to solve a wide array of different optimization problems, including most of the problems that can be solved using Genetic Algorithms (GA).

Many popular optimization algorithms are deterministic, such as gradient-based algorithms The PSO, similarly to the algorithms belonging to the Evolutionary Algorithm family, is a stochastic algorithm that does not need gradient information derived from the error function. This allows the PSO to be used on functions where the gradient is either unavailable or computationally expensive to obtain.

The origins of the PSO are best described as sociologically inspired since the original algorithm was based on the sociological behavior associated with bird flocking This topic will be discussed in more detail below after the basic algorithm has been described.

4.3 Optimizing Model with PSO Algorithm

PSO algorithm maintains a population of particles, where each particle represents a potential solution to an optimization problem. Let, *s* be the size of the swarm. Each particle *i* can be represented as an object with several characteristics. These characteristics are assigned the following symbols:

x_i: The current position of the particle

v_i: The current velocity of the particle

y_i: The personal best position of the particle

The personal best position associated with particle *i* is the best position that the particle has visited (a previous value of x), yielding the highest fitness value for that particle. For a minimization task, a position yielding a smaller function value is regarded as having a higher fitness. The symbol *f* is used to denote the objective function that is being minimized. The update equation for the personal best position is presented in equation (23), with the dependence on the time step t made explicit.

$$y\,(t+1)= \begin{cases} y_i(t) & \text{if } f\,(x_i(t+1)) \geq f(y_i\,(t)) \\ y_i(t+1) & \text{if } f\,(x_i(t+1) < f(y_i\,(t)) \end{cases} \tag{23}$$

Two versions of the PSO exist, called the Gbest and Lbest models. The difference between the two algorithms is based on the set of particles with which a given particle will interact with directly, where the symbol ŷ is used to represent this interaction. The details of the two models will be discussed in full below. The definition of ŷ, as used in the Gbest model, is presented in equation (24)

$$\hat{y}(t) \in \{y_0(t), y_1(t),\ldots\ldots, y_s(t)\} \;\Big|\; f(y_i\,(t)) = \min\,\{\, f(y_0\,(t)), f(y_1\,(t)),\ldots\ldots., f(y_s\,(t))\} \tag{24}$$

This definition states that ŷ is the best position discovered by any of the particles so far. The algorithm makes use of two independent random sequences, $r_1 \sim U\,(0, 1)$ and $r_2 \sim U\,(0, 1)$. These sequences are used to affect the stochastic nature of the algorithm, as shown below in equation (25). The values of r_1 and r_2 are scaled by constants $0 < c_1, c_2 \leq 2$. These constants are called the acceleration coefficients, and they influence the maximum size of the step that a particle can take in a single iteration. The velocity update step is specified separately for each dimension $j \in 1.....n$, so that $v_{i,j}$ denotes the j^{th} dimension of the velocity vector associated with the i^{th} particle . The velocity up date equation is then

$$v_{i,j}(t+1) = v_{i,j}(t) + c_1\, r_{1,j}(t)\, [y_{i,j}(t) - x_{i,j}(t)] + c_2\, r_{2,j}(t)[\hat{y}_j(t) - x_{i,j}(t)] \qquad (25)$$

From the definition of the velocity update equation it is clear that c_2 regulates the maximum step size in the direction of the global best particle, and c_1 regulates the step size in the direction of the personal best position of that particle. The value of $v_{i,j}$ is clamped to the range $[-v_{max}, v_{max}]$ to reduce the likelihood that the particle might leave the search space. If the search space is defined by the bounds $[-x_{max}, x_{max}]$, then the value of v_{max} is typically set so that $v_{max} = k \times x_{max}$, where $0.1 \leq k \leq 1.0$. The position of each particle is updated using the new velocity vector for that particle, so that

$$x_i(t+1) = x_i(t) + v_i(t+1) \qquad (26)$$

The algorithm consists of repeated application of the update equations presented above Figure 4.2 lists the pseudo- code for the basic PSO algorithm.

Create and initialize an n-dimensional PSO : S
Repeat:
  for each particle $i \in [1..s]$:
    if $f(S.x_i) < f(S.y_i)$
      then $S.y_i = S.x_i$
    if $f(S.y_i) < f(S.\hat{y})$
      then $S.\hat{y} = S.y_i$
endfor

Perform PSO updates on S using equations (5-9)
until stopping condition is true

Figure 4.2: Pseudo code for the original PSO algorithm

Here ***if*** statements are the equivalent of applying equations (23) and (24), respectively. The initialization mentioned in the first step of the algorithm consists the following:

1. Initialization of each coordinate of $x_{i,j}$ to a value drawn from the uniform random distribution on the interval [$-x_{max}$, x_{max}], for all $i \in 1....s$ and $j \in 1.....n$. This distributes the initial positions of the particles throughout the search space. Many of the pseudorandom number generators available have flaws leading to low-order correlations when used to generate random vectors this way, so care must be exercised in choosing a good pseudo-random algorithm. Alternatively, the initial positions can be distributed uniformly through the search space using sub-random sequences.

2. Initialization of each $v_{i,j}$ to a value drawn from the uniform random distribution on the interval [$-v_{max}$, v_{max}], for all $i \in 1....s$ and $j \in 1.....n$. Alternatively, the velocities of the particles could be initialized to 0, since the starting positions are already randomized.

3. Setting $y_i = x_i, \forall \in 1....s$. Alternatively, two random vectors can be generated for each particle, assigning the more fit vector to y_i and the less fit one to x_i. This would require additional function evaluations, so the simpler method described first is usually used.

4.4 PSO Flowchart

A brief description of how the algorithm works is shown in flowchart (Figure 4.3) and described as follows: Initially, some particle is identified as the best particle in a neighborhood of particles, based on its fitness. All the particles are then accelerated in the direction of this particle, but also in the direction of their own best solutions that they have discovered previously. Occasionally the particles will overshoot their target, exploring the search space beyond the current best particles. All particles also have the opportunity to discover better particles en route, in which case the other particles will change direction and head towards the new ‘best’ particle. Since most functions have some continuity, chances are that a good solution will be surrounded by equally good, or better, solutions. By approaching the current best solution from different directions in search space, the chances are good that these neighboring solutions will be discovered by some of the particles.

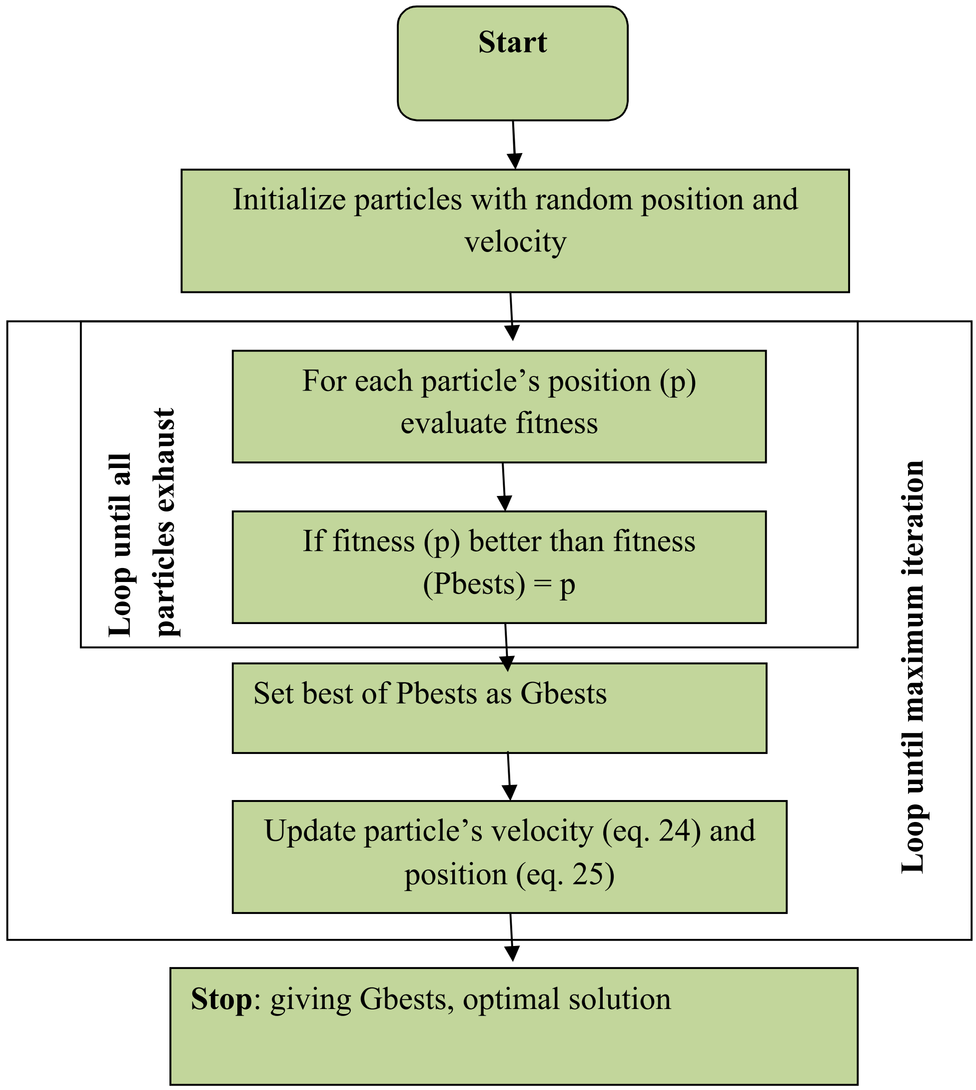

Figure 4.3: Basic PSO algorithm flowchart

The stopping criterion mentioned in Figure 4.3 depends on the type of problem being solved. Usually the algorithm is run for a fixed number of function evaluations (thus a fixed number of iterations) or until a specified error bound is reached. It is important to realize that the velocity term models the rate of change in the position of the particle. The changes induced by the velocity update equation (24) therefore represent acceleration, which explains why the constants c_1 and c_2 are called acceleration coefficients.

4.5 The Gbest model

The Gbest model offers a faster rate of convergence [39] at the expense of robustness. This model maintains only a single "best solution," called the global best particle, across all the particles in the swarm. This particle acts as an attractor, pulling all the particles towards it. Eventually all particles will converge to this position, so if it is not up dated regularly, the swarm may converge prematurely. The update equations for $\hat{y}_i$ and v_i are the ones presented above, repeated here for completeness.

$$\hat{y}(t) \in \{y_0(t), y_1(t), \ldots, y_s(t)\} \mid f(y_i(t)) = \min\{f(y_0(t)), f(y_1(t)), \ldots, f(y_s(t))\} \quad (27)$$

$$v_{i,j}(t+1) = v_{i,j}(t) + c_1 r_{1,j}(t)[y_{i,j}(t) - x_{i,j}(t)] + c_2 r_{2,j}(t)[\hat{y}_j(t) - x_{i,j}(t)] \quad (28)$$

Here ŷ is called the global best position, and belongs to the particle referred to as the global best particle.

4.6 The Lbest model

The Lbest model tries to prevent premature convergence by maintaining multiple attractors. A subset of particles is defined for each particle from which the local best particle, $\hat{y}_i$ is then selected. The symbol $\hat{y}_i$ is called the local best position or the neighborhood best. Assuming that the particle indices wrap around at *s*, the Lbest update equations for a neighborhood of size *l* are as follows

$$N_i = \{y_{i-l}(t), y_{i-l+1}(t), \ldots, y_{i-l}(t), y_i(t), y_{i+1}(t), \ldots, y_{i-l}(t), y_{i+l}(t)\} \quad (29)$$

$$\hat{y}_j(t+1) \in N_i \mid f(y_i(t+1)) = \min\{f(a)\}, \forall_a \in N_i \quad (30)$$

$$v_{i,j}(t+1) = v_{i,j}(t) + c_1 r_{1,j}(t)[y_{i,j}(t) - x_{i,j}(t)] + c_2 r_{2,j}(t)[\hat{y}_j(t) - x_{i,j}(t)] \quad (31)$$

Here it is noticeable that the particles selected to be in subset N_i have no relationship to each other in the search space domain; selection is based purely on the particle's index number. This is done for two main reasons: it is computationally inexpensive, since no clustering has to be performed, and it helps promote the spread of information regarding good solutions to all particles, regardless of their current location in search space.

Lastly, it is important to note that the gbest model is actually a special case of the lbest model with $l = s$. Experiments with $l = 1$ have shown the lbest algorithm to converge somewhat more slowly than the gbest version, but it is less likely to become trapped in an inferior local minimum [39].

4.7 Social Behavior of PSO

Many interpretations of the operation of the PSO have been suggested. Kennedy strengthened the socio-psychological view by performing experiments to investigate the function of the different components in the velocity update equation (24) [40]. Kennedy made use of the Lbest model rather than the Gbest model outlined above.

Consider the velocity update equation (24), repeated here for convenience.

$$v_{i,j}(t+1) = v_{i,j}(t) + c_1 \, r_{1,j}(t) \, [y_{i,j}(t) - x_{i,j}(t)] + c_2 \, r_{2,j}(t)[\hat{y}_j(t) - x_{i,j}(t)]$$

The term $c_1 \, r_{1,j}(t) \, [y_{i,j}(t) - x_{i,j}(t)]$ is associated with cognition since it only takes into account the particle's own experiences. If a PSO is constructed making use of the cognitive term only, the velocity update equation will become

$$v_{i,j}(t+1) = v_{i,j}(t) + c_1 \, r_{1,j}(t) \, [y_{i,j}(t) - x_{i,j}(t)] \quad (32)$$

Kennedy [40] found that the performance of this 'cognition only' model was inferior to that of the original swarm, failing to train the network within the maximum allowed number of iterations for some parameter settings. One of the reasons for the poor behavior of this version of the PSO is that there is no interaction between the different particles.

The third term in the velocity update equation, $c_2 r_{2,j}(t)[\hat{y}_j(t) - x_{i,j}(t)]$ represents the social interaction between the particles. A 'social only' version of the PSO can be constructed by using the following velocity update equation.

$$v_{i,j}(t+1) = v_{i,j}(t) + c_2 \, r_{2,j}(t)[\hat{y}_j(t) - x_{i,j}(t)] \quad (33)$$

The performance of this model was superior to that of the original PSO on the specific problem that Kennedy investigated.

In summary, the PSO velocity update term consists of both a cognition component and a social component. Little is currently known about the relative importance of these two terms, although initial results seem to indicate that the social component may be more significant on some problems.

4.8 Comparison with other Evolutionary Algorithms

The PSO is clearly related to some of the evolutionary algorithms. For one, the PSO maintains a population of individuals representing potential solutions, a property common to all Evolutionary Algorithms (EA). If the personal best positions (y_i) are treated as part of the population, then there is clearly a weak form of selection. In a ($\mu + \lambda$) Evolutionary Search (ES) algorithm [41], the offspring compete with the parents, replacing them if they are more fit. The update equation (24) resembles this mechanism, with the difference that each personal best position (parent) can only be replaced by its own current position (offspring), should the current position be more fit than the old personal best position. To summarize, there appears to be some weak form of selection present in the PSO.

The velocity update equation resembles the arithmetic crossover operator found in real valued GAs. Normally, the arithmetic crossover produces two offspring that are linear blends of the two parents involved. The PSO velocity update equation, without the $v_{i,j}$ (t) term (equation 25), can be interpreted as a form of arithmetic crossover involving two parents , returning a single offspring. Alternatively, the velocity update equation, without the $v_{i,j}(t)$ term, can be seen as a mutation operator, with the strength of the mutation governed by the distance that the particle from its two 'parents'. This still leaves the $v_{i,j}(t)$ term uncounted for, which can be interpreted as a form of mutation dependent on the position of the individual in the previous iteration.

A better way of modeling the $v_{i,j}(t)$ terms to think of each iteration not as a process of replacing the previous population with a new one (death and birth), but rather as a process of adaption [42]. This way the $x_{i,j}$ values are not replaced, but rather adapted using the velocity vectors v_i. This makes the difference between the other EAs and the PSO more conspicuous: the PSO maintains information regarding position and velocity (changes in position); in contrast, traditional EAs only keep track of positions. Therefore it appears that there is some degree of overlap between the PSO and most other EAs, but the PSO has some characteristics that are currently not present in other EAs, especially the fact that the PSO models the velocity of the particles as well as the positions.

4.9 PSO Origins and Terminology

The movement of the particles has been described as "flying" through n-dimensional space [39]. This terminology is in part due to experiments with bird flocking simulations which led to the development of the original PSO algorithm [40]. Several reasons have been found for the flocking behavior observed in nature. Some evolutionary advantages include: protection from predators, improved survival of the gene pool, and profiting from a larger effective search area with respect to food. This last property is invaluable when the food is unevenly distributed over a large region.

Reynolds [39] proceeded to model his flocks using three simple rules: collision avoidance, velocity matching and flock centering. The flock centering drive will prompt a bird to fly closer to its neighbors (so that the velocity matching is not jeopardized) but still maintaining a safe distance, as governed by the collision avoidance rule. Reynolds decided to use a flock centering drive calculated by considering only the nearest neighbors of a bird, instead of using the centre of the whole swarm, which he called the "central force model". This corresponds roughly to the lbest model of the PSO.

Even though the particle movement visually looks like flocking, it does not strictly comply with certain definitions of flocking behavior. Mataric defines the following concepts flock behavior [43]:

i. **Safe-Wandering:** The ability of a group of agents to move about while avoiding collisions with obstacles and each other.
ii. **Dispersion:** The ability of a group of agents to spread out in order to establish and maintain some minimum inter-agent distance.

iii. **Aggregation:** The ability of a group of agents to gather in order to establish and maintain some maximum inter- agent distance.

iv. **Homing:** The ability to find a particular region or location

The PSO only implements homing and aggregation, lacking safe wandering and dispersion. Safe-wandering is not important to the PSO, since it only applies to entities that can collide physically. Dispersion means that the particles will fan out when they get too close to one another, something which is not currently in the PSO model - the PSO encourages the particles to cluster. The terms “swarm” and “swarming” are much more appropriate. This term was used by Millonas to describe artificial life models [44]. Millonas suggested that swarm intelligence is characterized by the following properties:

i. **Proximity**: Carrying out simple space and time computations.

ii. **Quality**: Responding to quality factors in the environment.

iii. **Diverse Response**: Not falling into a restricted subset of solutions.

iv. **Stability**: Being able to maintain mo des of behavior w hen the environment changes.

v. **Adaptability**: Being able to change behavioral modes when deemed profitable.

Particles in the PSO possess these properties. The term “particle” has some justification. The members of the population lack mass and volume, thus calling them “points” would be more accurate. The concepts of velocity and acceleration, however, are more compatible with the term particle (alluding to a small piece of matter) than they are with the term point. Some other research fields, notably computer graphics, also use the term “particle systems” to describe the models used for rendering effects like smoke or fire

4.10 Application of PSO

The PSO has been applied to a vast number of problems, though not all of these applications have been described in published material yet. This section will briefly describe some of the applications of PSO.

Neural network training was one of the first applications of the PSO. Kennedy and Eberhart [36] reported that the PSO was successful in training a network to correctly classify the XOR problem, a process involving the minimization of function in a 13-dimensional search space. They also reported that the PSO could train a neural network to classify Fisher's Iris Data set although few details were provided. Salerno also applied the PSO to the task of training a neural network to learn the XOR problem, reporting significantly better performance than that obtained with a Gradient Descent algorithm [45]. He also showed that the PSO was able to train a simple recurrent neural network.

In fact, most PSO applications reported in the literature involve neural network training. Earlier versions of the PSO, before the introduction of the inertia weight or constriction factor, did not have the ability to perform a fine-grained search of the error surface. This leads to experiments involving a hybrid between PSO and traditional gradient techniques. Van den Bergh used the PSO to find a suitable starting position for the Scaled Conjugate Gradient algorithm [46]. Results showed that the hybrid method resulted in significantly better performance on both classification problems, using the UCI Ionosphere problem [45] as example, and function approximation problems, using the he non-curve time series as example.

Eberhart et al. [42] used the PSO to train a network to correctly classify a patient as exhibiting essential tremor, or suffering from Parkinson's disease. Their PSO implementation used an inertia weight that decreased linearly from 0.9 to 0.4 over 2000 iterations. An interesting feature of the neural network the y used was that they trained the slop e of the sigmoidal activation functions along with the weights of the network.

The PSO has also been used to evolve the architecture of a neural network in tandem with the training of the weights of the network. Zhang and Shao report that their PSONN algorithm [47] was able to successfully evolve a network for estimating the quality of jet fuel. Part of the P SONN system involve s the optimization of the number of hidden units used in the network — a task that is handled by a PSO. Whenever a new hidden node is added to the network, only the newly added nodes are trained (again using a PSO) in an attempt to reduce the error, greatly improving the speed of the algorithm.

Eberhart et al. describe several other applications of the PSO [40], including some more neural network training applications. Tandon used the PSO to train a neural network used to simulate and control an end milling process [48]. End milling involves the removal of metal in a manufacturing process using computer numerically controlled (CNC) machine tools. Another neural network training application described by Eberhart et al. is that of training a network to estimate the state -of-charge of a battery pack.

Yet another neural network training application, this time using a Fuzzy Neural Network, was studied by He et al. [49]. One of the more interesting points regarding their research was the fact that they modified the velocity update equation so that it is no longer accumulative, i.e.

$$v_{i,j}(t+1) = c_1\, r_{1,j}(t)\, [y_{i,j}(t) - x_{i,j}(t)] + c_2\, r_{2,j}(t)[\hat{y}_j(t) - x_{i,j}(t)] \qquad (34)$$

Results are presented showing that this modified velocity up date equation gave rise to improved performance on some benchmark functions. Note that this modification makes their version of the PSO very similar to an Evolution Strategies algorithm. They also showed that their fuzzy neural network was able to produce a set of 15 rules with a classification accuracy of 97% on the Iris data set. This is a rather large number of rules for such a simple classification task, compared to other efficient algorithms.

The PSO was used to optimize the ingredient mix of chemicals used to facilitate the growth of strains of micro organisms, resulting in a significant improvement over the solutions found by previous optimization methods. One of the strengths of the PSO is the ability to explore a large area of search space relatively efficiently; this property led the PSO to discover a better solution in a location in search space very different from the solutions discovered by other existing techniques.

Another application unrelated to neural network training was published by Fukuyama and Yoshida [50]. They have shown that the PSO is very effective at optimizing both continuous and discrete variables simultaneously. The PSO velocity update equation can be adapted for use with discrete variables by simply discretising the values before using them in the velocity update step (using for example equation 25). The position of the particle is also discretised after equation (26) has been used to update it. These discrete variables can be mixed freely with the continuous variables, as long as the appropriate (discretised or continuous) update equations are applied to them. The application on which Fukuyama and Yoshida demonstrated their modified PSO was that of calculating the correct response to changes in the load on an electric power grid. This problem requires the simultaneous optimization of numerous discrete and continuous variables, and was traditionally solved using the Reactive Tabu Search (RTS) algorithm. The RTS algorithm scales very poorly with problem dimensionality, since the number of candidates for evaluation increases exponentially with the dimension of the problem. To illustrate, the RTS algorithm required 7.6 hours to solve a problem consisting of 1217 busses. The same problem was solved in only 230 seconds using a PSO with a linearly decreasing inertia weight in the range [0.9, 0.4].

4.11 Reasons to Choose PSO

PSO is a derivative-free algorithm unlike many conventional techniques. It has the flexibility to be integrated with other optimization techniques to form a hybrid tool. It is less sensitive to the nature of the objective function, i.e. convexity or continuity. It has less parameter to adjust unlike many other competing evolutionary techniques. It has the ability to escape local minima. Another point is that, PSO is easy to implement and program with basic mathematical and logic operations. It can handle objective functions with stochastic nature. Moreover, it does not require a good initial solution to start its iteration process. All these reasons make PSO a strong candidate to use as meta-heuristic algorithm to search the optimal solution.

Though a lot of applications have been done in PSO within a short time, so far PSO has not directly applied in solving EPQ model. The PSO code developed for the proposed model is given in Appendix I.

CHAPTER 5
INVENTORY MODELING WITH GENETIC ALGORITHM (GA)

In order to show that the proposed PSO algorithm is an effective means of solving the complicated inventory model of this research, the Genetic Algorithm (GA) is also employed to optimize the model. GA is well known as a search algorithm and has been widely used in different field of study. GA has been also used for inventory optimization problem by many researchers. In the literature section it has been discussed. In this section definition, basic idea and components of GA will be provided.

GA is an iterative procedure maintaining a population of structures that are candidate solutions to specific domain challenges. During each temporal increment (called a generation), the structures in the current population are rated for their effectiveness as domain solutions, and on the basis of these evaluations, a new population of candidate solutions is formed using specific *genetic operators* such as reproduction, crossover, and mutation.

GA was developed by John Holland in 1975. It is a search algorithm based on the mechanics of the natural selection process (biological evolution). The most basic concept is that the strong tend to adapt and survive while the weak tend to die out. That is, optimization is based on evolution, and the "Survival of the fittest" concept. GAs has the ability to create an initial population of feasible solutions, and then recombine them in a way to guide their search to only the most promising areas of the state space. Each feasible solution is encoded as a chromosome (string) also called a genotype, and each chromosome is given a measure of fitness via a fitness (evaluation or objective) function. The fitness of a chromosome determines its ability to survive and produce offspring. A finite population of chromosomes is maintained. GAs use probabilistic rules to evolve a population from one generation to the next. It is a robust search technique and produce "close" to optimal results in a "reasonable" amount of time.

5.1 General Scheme of GAs

The general scheme of a GA is shown in figure 5.1.

```
begin
   INITIALIZE population with random candidate solutions;
   EVALUATE each candidate;
   repeat
      SELECT parents;
      RECOMBINE pairs of parents;
      MUTATE the resulting children;
      EVALUATE children;
      SELECT individuals for the next generation

   until TERMINATION-CONDITION is satisfied
```

Figure 5.1: The general scheme of Genetic Algorithm

The GA can be represented in form of a diagram also:

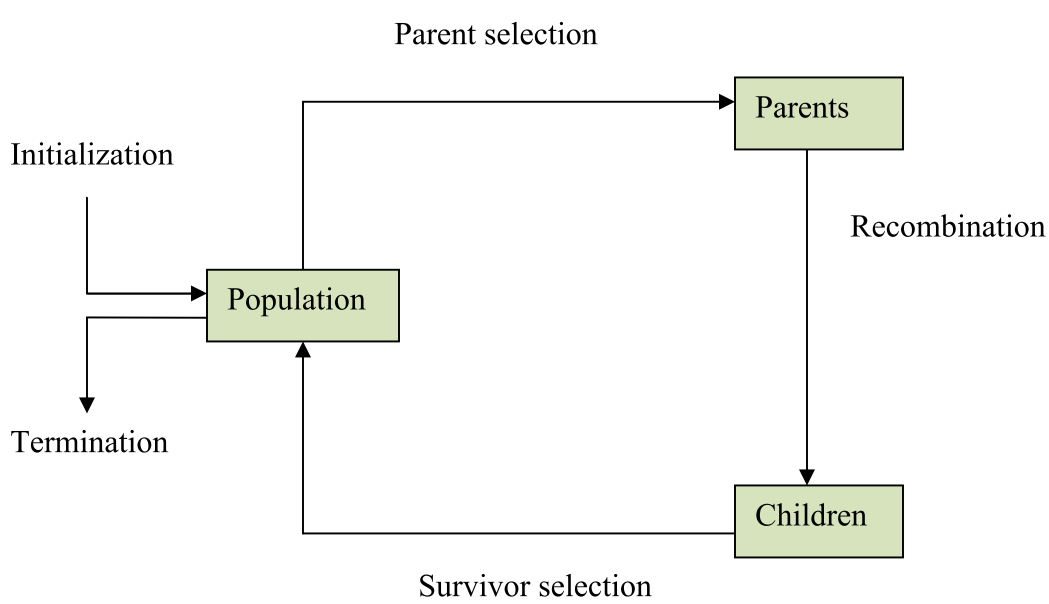

Figure 5.2: Genetic Algorithm Diagram

GA has a number of features:

i. GA is population-based
ii. GA uses recombination to mix information of candidate solutions into a new one.
iii. GA is stochastic.

It's clear that this scheme falls in the category of *generate-and-test* algorithms. The evaluation function represents a heuristic estimation of solution quality and the search process is driven by the variation and the selection operator.

5.2 Components of GAs

The most important components in a GA consist of:

i. Representation (definition of individuals)
ii. Evaluation function (or fitness function)
iii. Population
iv. Parent selection mechanism
v. Variation operators (crossover and mutation)
vi. Survivor selection mechanism (replacement)

5.2.1 Representation

Objects forming possible solution within original problem context are called *phenotypes*, their encoding, the individuals within the GA, are called *genotypes*. *Candidate solution*, *phenotype* and *individual* are used to denote points of the space of possible solutions. This space is called *phenotype space*. *Chromosome* and *individual* can be used for points in the *genotype space*. Elements of a chromosome are called *genes*.

5.2.2 Mutation Operator

A unary variation operator is called *mutation*. It is applied to one genotype and delivers a modified *mutant*, the *child* or *offspring* of it. In general, mutation is supposed to cause a random unbiased change. Mutation has a theoretical role: it can guarantee that the space is connected.

Example: Assume that we have already used crossover to get a new string: 7 3 4 5 1 3. Assume the mutation rate is 0.001 (usually a small value). Next, for the first bit 7, we generate randomly a number between 0 and 1. If the number is less than the mutation rate (0.001), then the first bit 7 needs to mutate. We generate another number between 1 and the maximum value 8, and get a number (for example 2). Now the first bit mutates to 2. We repeat the same procedure for the other bits. In our example, if only the first bit mutates, and the rest of the bits don't mutate, then we will get a new chromosome as below:

2 3 4 5 1 3

5.2.3 Crossover Operator

A binary variation operator is called *recombination* or *crossover.* This operator merges information from two parent genotypes into one or two offspring genotypes. Similarly to mutation, crossover is a stochastic operator: the choice of what parts of each parent are combined and the way these parts are combined depend on random drawings. The principle behind crossover is simple: by mating two individuals with different but desirable features, an offspring can be produce which combines both of those features. There are many kinds of crossover.

5.2.3.1 One-point crossover

The procedure of one-point crossover is to randomly generate a number (less than or equal to the chromosome length) as the crossover position. Then, keep the bits before the number unchanged and swap the bits after the crossover position between the two parents.

Example: With the two parents selected above, randomly a number 2 as the crossover position is generated:

Parent1: 7 3 7 6 1 3
Parent2: 1 7 4 5 2 2

Then we get two children:

Child 1: 7 3| 4 5 2 2

Child 2: 1 7| 7 6 1 3

5.2.3.2 Two-point cross Over

The procedure of two-point crossover is similar to that of one-point crossover except that we must select two positions and only the bits between the two positions are swapped. This crossover method can preserve the first and the last parts of a chromosome and just swap the middle part.

Example: With the two parents selected above, randomly two numbers 2 and 4 as the crossover positions is generated:

Parent1: 7 3 7 6 1 3

Parent2: 1 7 4 5 2 2

Then we get two children: Child 1 : 7 3| 4 5| 1 3

Child 2 : 1 7| 7 6| 2 2

5.2.3.3 Uniform Crossover

The procedure of uniform crossover, each gene of the first parent has a 0.5 probability of swapping with the corresponding gene of the second parent.

Example: For each position, we randomly generate a number between 0 and 1, for example, 0.2, 0.7, 0.9, 0.4, 0.6, 0.1. If the number generated for a given position is less than 0.5, then child1 gets the gene from parent1, and child2 gets the gene from parent2. Otherwise, vice versa.

Parent1: 7 *3 *7 6 *1 3

Parent2: 1 *7 *4 5 *2 2

Then we get two children:

Child 1 : 7 7* 4* 6 2* 3

Child 2 : 1 3* 7* 5 1* 2

5.3 Inversion

Inversion operates as a kind of reordering technique. It operates on a single chromosome and inverts the order of the elements between two randomly chosen points on the chromosome. While this operator was inspired by a biological process, it requires additional overhead.

Example: Given a chromosome 3 8 4 8 6 7. If randomly two positions 2, 5 are chosen and apply the inversion operator, then the new string: 3 6 8 4 8 7.

5.4 Parent Selection Mechanism

The role of *parent selection* (*mating selection*) is to distinguish among individuals based on their quality to allow the better individuals to become parents of the next generation. Parent selection is *probabilistic*. Thus, high quality individuals get a higher chance to become parents than those with low quality. Nevertheless, low quality individuals are often given a small but positive chance; otherwise the whole search could become too greedy and get stuck in a local optimum.

The standard, original method for parent selection is Roulette Wheel selection or fitness-based selection. In this kind of parent selection, each chromosome has a chance of selection that is directly proportional to its fitness. The effect of this depends on the range of fitness values in the current population.

Example: if fitness range from 5 to 10, then the fittest chromosome is twice as likely to be selected as a parent than the least fit. If we apply fitness-based selection on the population given in example 3.1, we select the second chromosome 7 3 7 6 1 3 as our first parent and 1 7 4 5 2 2 as our second parent.

There are also other types of selection mechanisms. In the rank-based selection method, selection probabilities are based on a chromosome's relative rank or position in the population, rather than absolute fitness. The original tournament selection is to choose K parents at random and returns the fittest one of these.

5.5 Survivor Selection Mechanism

The role of survivor selection is to distinguish among individuals based on their quality. In GA, the population size is (almost always) constant, thus a choice has to be made on which individuals will be allowed in the next generation. This decision is based on their fitness values, favoring those with higher quality. As opposed to parent selection which is stochastic, survivor selection is often *deterministic*, for instance, ranking the unified multiset of parents and offspring and selecting the top segment (fitness biased), or selection only from the offspring (age-biased).

5.6 Initialization and Termination Condition

Initialization is kept simple in most GA applications. Whether this step is worth the extra computational effort or not is very much depending on the application at hand. GA is stochastic and mostly there are no guarantees to reach an optimum. Commonly used conditions for terminations are the following:

i. The maximally allowed CPU times elapses
ii. The total number of fitness evaluations reaches a given limit
iii. For a given period of time, the fitness improvement remains under a threshold value
iv. The population diversity drops under a given threshold.

5.7 Constraint Handling in GAs

There are many ways to handle constraints in a GA. At the high conceptual level we can distinguish two cases: indirect constraint handling and direct constraint handling. Indirect constraint handling means that we circumvent the problem of satisfying constraints by incorporating them in the fitness function f such that f optimal implies that the constraints are satisfied, and use the power of GA to find a solution. Direct constraint handling means that we leave the constraints as they are and 'adapt' the GA to enforce them. Direct and indirect constraint handling can be applied in combination, i.e., in one application we can handle some constraints directly and others indirectly. Formally, indirect constraint handling means transforming constraints into optimization objectives.

5.7.1 Direct constraint handling

Treating constraints directly implies that violating them is not reflected in the fitness function, thus there is no bias towards chromosomes satisfying them. Therefore, the population will not become less and less infeasible with respect to these constraints. This means that we have to create and maintains feasible chromosomes in the population. The basic problem in this case is that the regular operators are blind to constraints, mutating one or crossing over two feasible chromosomes can result in infeasible offspring. Typical approaches to handle constraints directly are the following:

i. Eliminating infeasible candidates
ii. Repairing infeasible candidates
iii. Preserving feasibility by special operators
iv. Decoding, i.e. transforming the search space.

Eliminating infeasible candidates is very inefficient, and therefore hardly applicable. Repairing infeasible candidates requires a repair procedure that modifies a given chromosome such that it will not violate constraints. This technique is thus problem dependent. The preserving approach amounts to designing and applying problem-specific operators that do preserve the feasibility of parent chromosomes. Note that the preserving approach requires the creation of a feasible initial population, which can be NP-complete. Decoding can simplify the problem search space and allow an efficient genetic algorithm. Formally, decoding can be seen as shifting to a search space that is different from the Cartesian product of the domains of the variables in the original problem formulation.

5.7.2 Indirect constraint handling

In the case of indirect constraint handling the optimization objectives replacing the constraints are viewed *penalties* for constraint violation hence to be minimized. In general penalties are given for violated constraints although some GAs allocates penalties for wrongly instantiated variables or as the distance to a feasible solution. Advantages of indirect constraint handling are:

i. Generality

ii. Reduction of the problem to 'simple' optimization
iii. Possibility of embedding user preferences by means of weights.

Disadvantages of indirect constraint handling are:

i. Loss of information by packing everything in a single number
ii. Does not work well with sparse problems.

5.8 When to Use GAs?

GA is generally used in following cases-

i. When an acceptable solution representation is available
ii. When a good fitness function is available
iii. When it is feasible to evaluate each potential solution
iv. When a near-optimal, but not optimal solution is acceptable.
v. When the state-space is too large for other methods

5.9 Applications of GAs

i. Scheduling: Facility, Production, Job, and Transportation Scheduling
ii. Design: Circuit board layout, Communication Network design, keyboard layout, Parametric design in aircraft
iii. Control: Missile evasion, Gas pipeline control, Pole balancing
iv. Machine Learning: Designing Neural Networks, Classifier Systems, Learning rules
v. Robotics: Trajectory Planning, Path planning
vi. Combinatorial Optimization: TSP, Bin Packing, Set Covering, Graph Bisection, Routing,
vii. Signal Processing: Filter Design
viii. Image Processing: Pattern recognition
ix. Business: Economic Forecasting; Evaluating credit risks, Detecting stolen credit cards before customer reports it is stolen
x. Medical: Studying health risks for a population exposed to toxins

CHAPTER 6
RESULTS AND DISCUSSIONS

In this thesis an economic production inventory model is developed considering some practical situations such as 100% inspection, defective products and reliability of the production process. The production inventory model is composed of production cost, holding cost, setup cost, inspection cost, depreciation cost, interest cost, rejection cost and backorder cost. The unconstrained non integer non linear model is used to determine the optimal values of different decision variables i.e. lot size per cycle, reliability of the production process and duration of time until the production is being held. The model is discussed by illustrating a numerical example.

6.1 Numerical Illustration

Three EPQ problems have been considered to illustrate the model. For the three different EPQ problems following different dataset are considered:

Table 6.1: Data for three EPQ problem

Parameters	EPQ Problem 1	EPQ Problem 2	EPQ Problem 3
C	50	100	10
D	20	10	100
P	30	20	120
h	1.5	1	2
S	250	300	500
l	800	1200	1600
m	0.5	0.4	0.6
n	0.9	0.8	0.6
I	0.1	0.1	0.01
j	0.1	0.1	0.01
α	0.4	0.05	0.1
β	0.8	0.05	0.1

6.1.1 Optimization of the models using PSO

In this study, the problem is solved by particle swarm optimization algorithm. The PSO code is developed in MATLAB. The code is attached in appendix. An important issue in PSO algorithm is to select its parameters. From the study of Kennedy [40] it is found that PSO works well if number of population varies from 30 to 50, inertia weight (k) ranges from 0.2 to 0.9, acceleration constants (c_1, c_2) vary from 1 to 4. Table 6.2 shows the different values of the PSO parameters used to obtain the solution.

Table 6.2: PSO Parameters for EPQ problems

Parameters	EPQ Problem 1	EPQ Problem 2	EPQ Problem 3
Inertia weight (k)	0.2	0.4	0.4
Acceleration constant (c1)	2.8	2	3.2
Acceleration constant (c2)	1.6	1.8	2.0
Swarm size (n)	50	40	30
Maximum iteration (i)	500	500	500

In any optimization another important issue is the upper and lower bound of decision variables. In our case, reliability (r) can vary from to 0 to 100%. Cycle time (T) and Lot size (Q) can't be negative. Table 6.3 shows the optimum results obtained from PSO.

Table 6.3: Results of PSO algorithm

EPQ Problems	Production process reliability (r)	Cycle Time (T)	Lot size (Q)	Total Cost per cycle
EPQ Problem 1	0.987	1.6957	49.814	1137.526
EPQ Problem 2	0.978	2.3808	40.249	1111.521
EPQ Problem 3	0.994	1.66	150.336	1492.922

From the results, it is found that reliability never reaches to 1.00. With the increase of reliability, holding cost increases but all other cost decreases. As a result reliability reaches to an optimum value then decreases in order to minimize the cost. All other results found are also reasonable. Convergence path of the objective function by PSO is shown in Figure 6.1 to Figure 6.3 for EPQ problems 1 to 3.

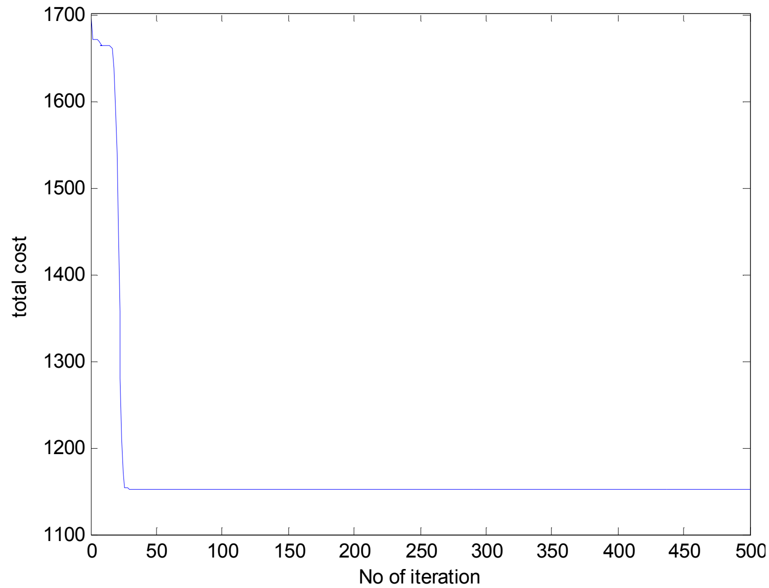

Figure 6.1: Convergence path of the objective function by PSO (Problem 1)

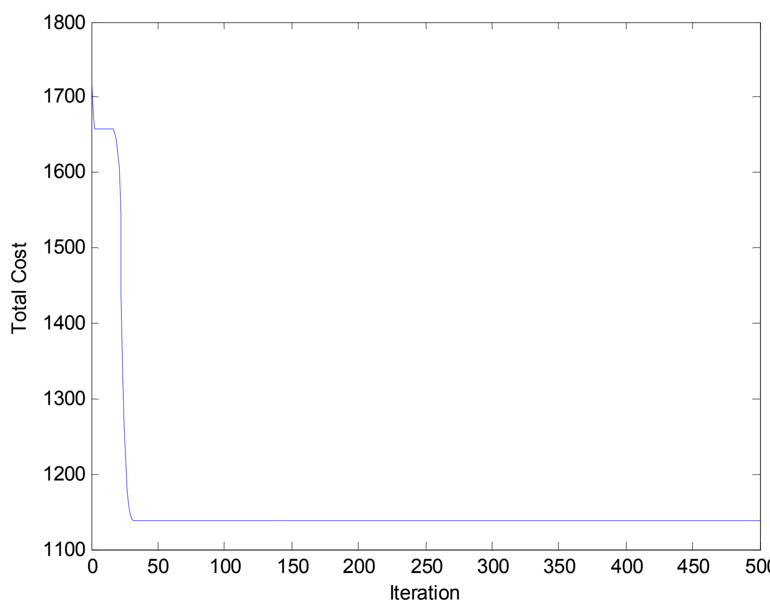

Figure 6.2: Convergence path of the objective function by PSO (Problem 2)

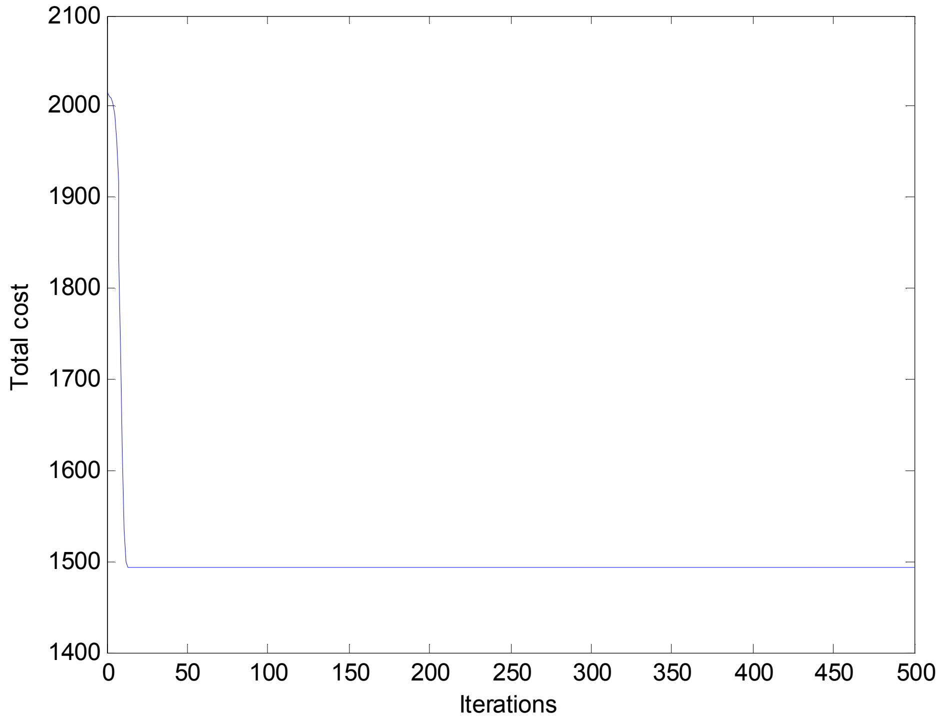

Figure 6.3: Convergence path of the objective function by PSO (Problem 3)

6.1.2 Optimization of the model using GA

So far developed, PSO has been used to solve numerous problems, but this is the first application where PSO is used directly to solve an EPQ model. GA has been widely used in various applications and it was previously applied in EPQ model several times. So, in order to validate the result obtained from PSO we run Genetic Algorithm (GA) simulation in the similar problem. The GA simulation is also run in MATLAB. In this case the following GA parameters are used in MATLAB.

Population type = Double vector

Population size =30

Scaling function= rank

Crossover = scattered

Crossover fraction = 0.8

Mutation function = constraint dependant

Migration =forward

Maximum generation = 200/500 (stopping criteria)

Function tolerance = 1×10^{-6}

Results obtained from GA are shown in Table 6.4. Convergence path of the objective function by PSO is shown in Figure 6.4 to Figure 6.6 for EPQ problems 1 to 3.

Table 6.4: Results of GA

EPQ Problems	Production process reliability (r)	Cycle Time (T)	Lot size (Q)	Total Cost per cycle
EPQ Problem 1	0.984	1.3422	32.9179	1247.559
EPQ Problem 2	0.993	2.1612	36.3662	1193.0534
EPQ Problem 3	0.987	1.7029	158.4336	1546.4525

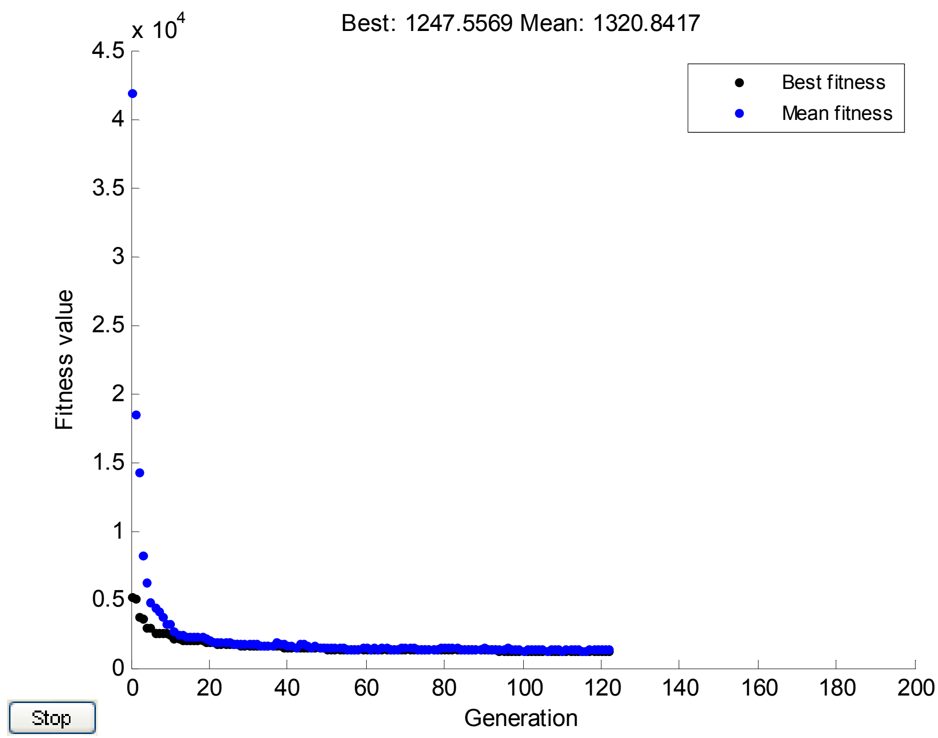

Figure 6.4: Convergence path of the objective function by GA (Problem 1)

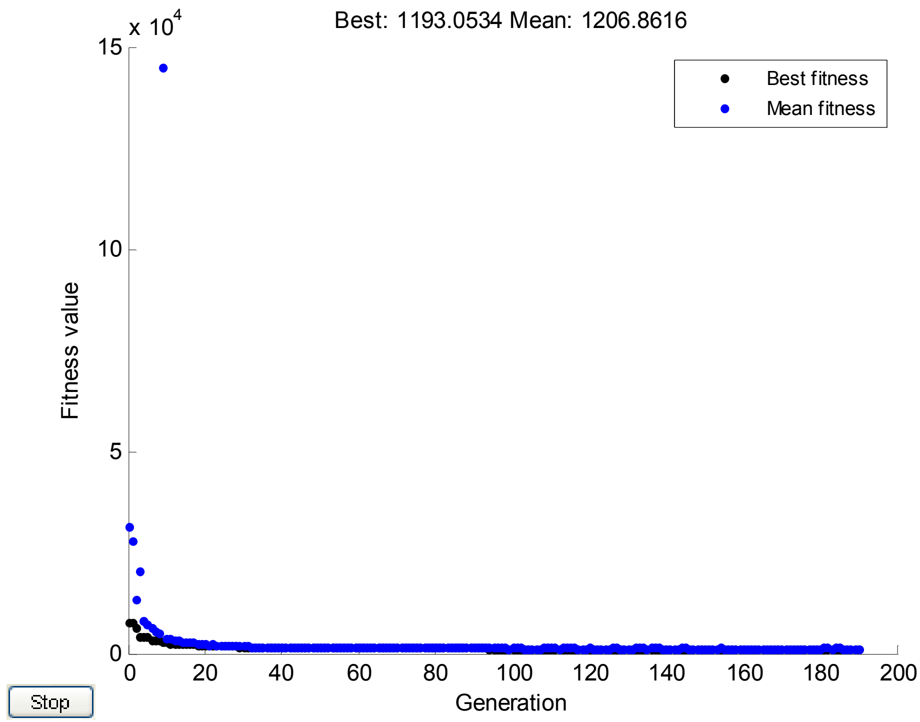

Figure 6.5: Convergence path of the objective function by GA (Problem 2)

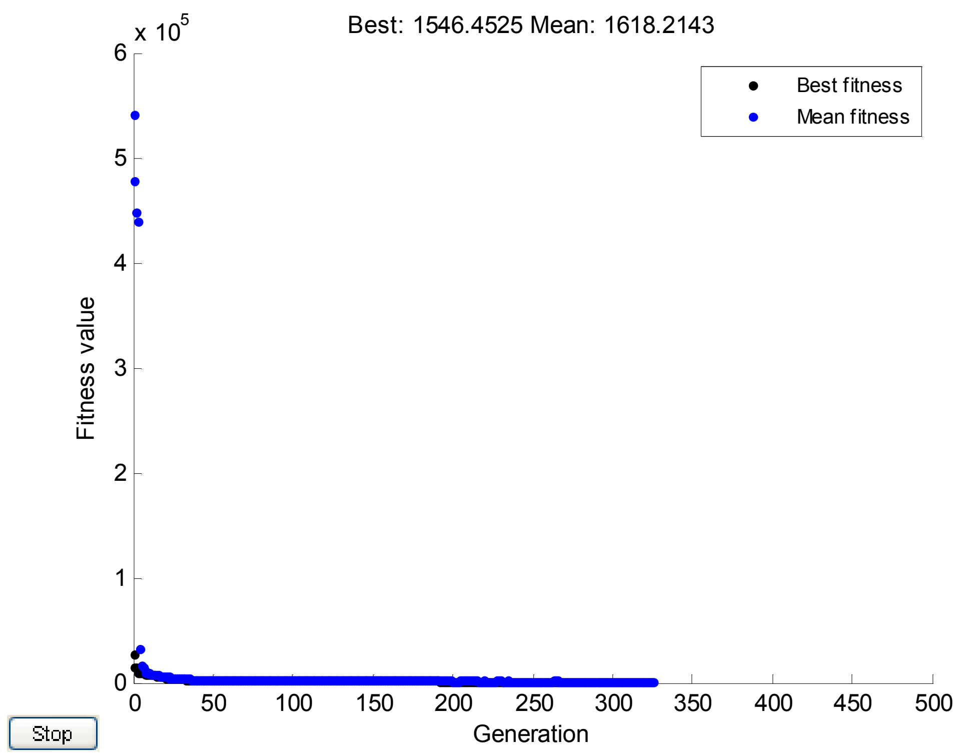

Figure 6.6: Convergence path of the objective function by GA (Problem 3)

6.2 Comparison of the Results of PSO and GA

From the Table 6.5 it is found that the results obtained from PSO is in good agreement with GA. In all three cases PSO outperforms GA in minimizing total cost per cycle. This is because PSO does not get stacked in local minima, has less parameter to GA, does not require a good initial solution and is less sensitive to the nature of the objective function than GA.

Table 6.5: PSO and GA comparative results

EPQ Problems	**r**		**T**		**Q**		**TC**	
	PSO	**GA**	**PSO**	**GA**	**PSO**	**GA**	**PSO**	**GA**
1	0.987	0.984	1.69	1.34	49.814	32.9179	1137.526	1247.559
2	0.978	0.993	2.38	2.16	40.249	36.3662	1111.521	1193.053
3	0.994	0.989	1.66	1.7	150.336	158.433	1492.922	1546.452

Table 6.6 gives the required number of generations, which are required to get the minimum total cost per cycle. From the table 6.6 we found that PSO requires less number of generations than GA to rich the optimum solution. Its means PSO requires less computational time than GA.

Table 6.6: No of generations required to reach minimum total cost for PSO and GA

EPQ Problems	PSO(number of iterations)	GA (number of generations)
1	30	110
2	35	120
3	15	50

The code has been developed in MATLAB 7.60 version with core-2-duo a 3.2 GHz processor and 2 GB ram personal computer. It has been found that, in PSO, for the increases of population size 30 to 1000, objective function improvement was negligible. It means PSO is relatively less sensitive to population size.

6.3 Relationship Analysis

Three decision variables- cycle time (T), Lot size (Q) and reliability of the production process (r) are related to each other. Figure 6.7 shows the relationship between production period/Cycle time (T) and reliability (r). With improvement of reliability of the system, Cycle time is decreased. In the study of Tripathy (2011), it is shown that production period and reliability are inversely related. Similar result also found in our case.

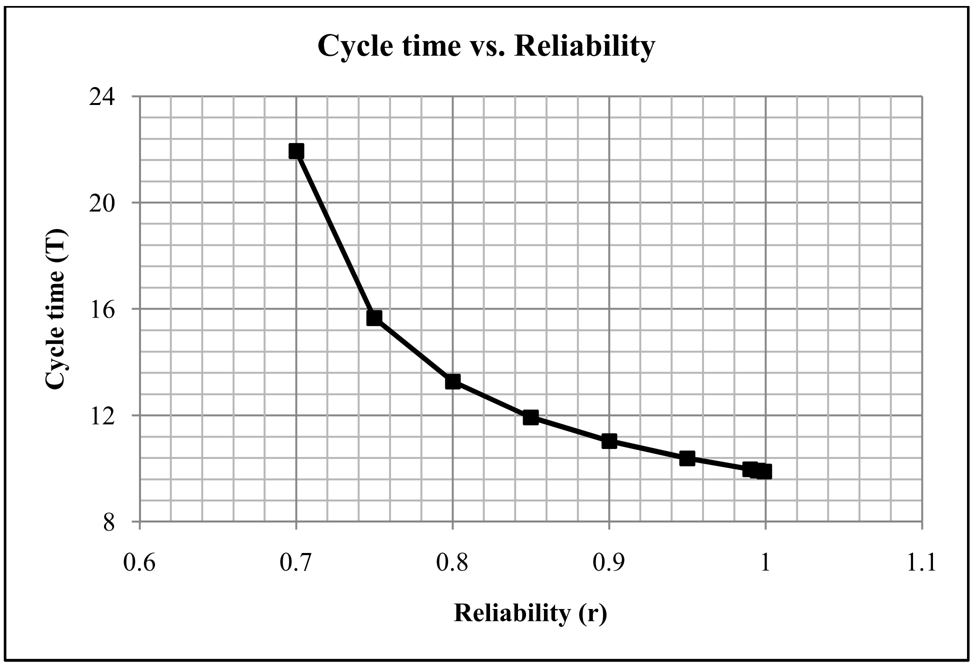

Figure 6.7: Relationship between cycle time and reliability

Production lot size (Q) and Production cycle time (T) are also related to each other. It is known for the traditional EPQ model that Q is proportional to cycle time (T) and demand. That is, with the increase of cycle time (T) lot size (Q) also increases. The relationship between set-up cost and reliability is shown in Figure 6.8

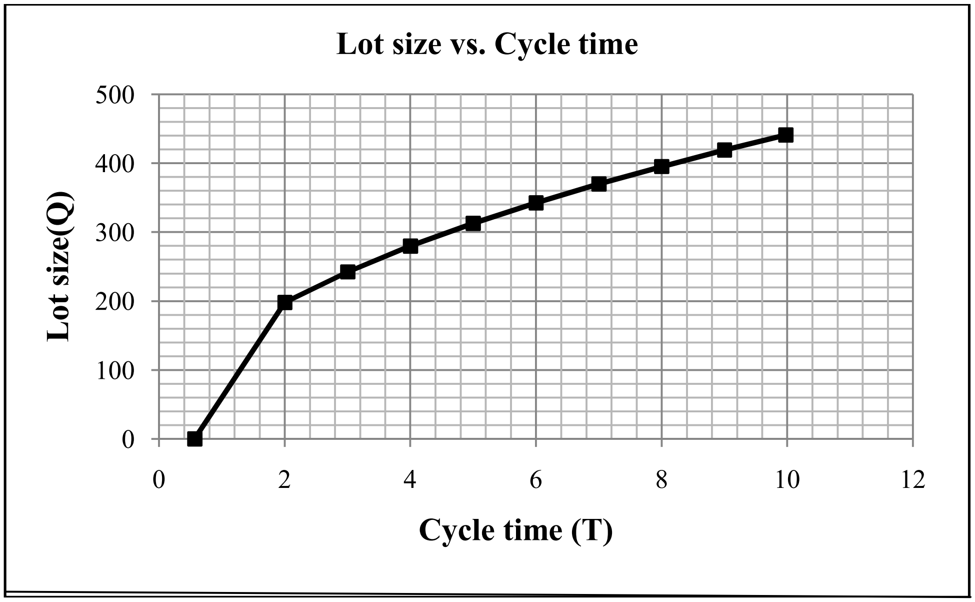

Figure 6.8: Relationship between lot size and reliability

Total cost is effected by three decision variables namely reliability of the production process, production period and economic lot size. Figure 6.9 shows the relationship between total cost and reliability. Total cost decreases with improvement of reliability up to 0.987. Total cost goes upward when reliability is greater than 0.987.

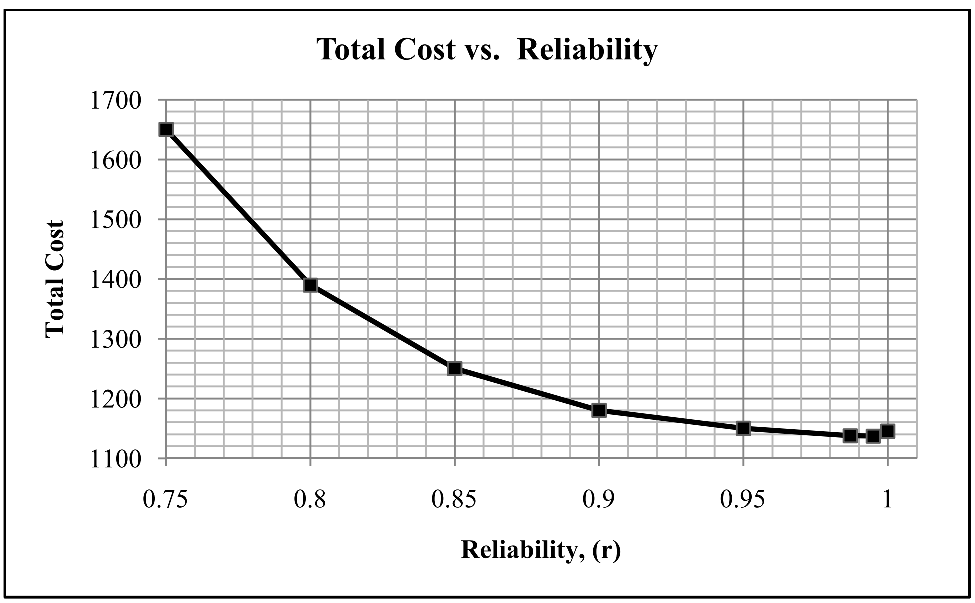

Figure 6.9: Relationship between total cost and reliability

From figure 6.10 it is found that total cost is also affected by cycle time (T). That is, with the increases of cycle time total cost decreases. This is because, when cycle time increases, we require less number of cycle and less setup cost.

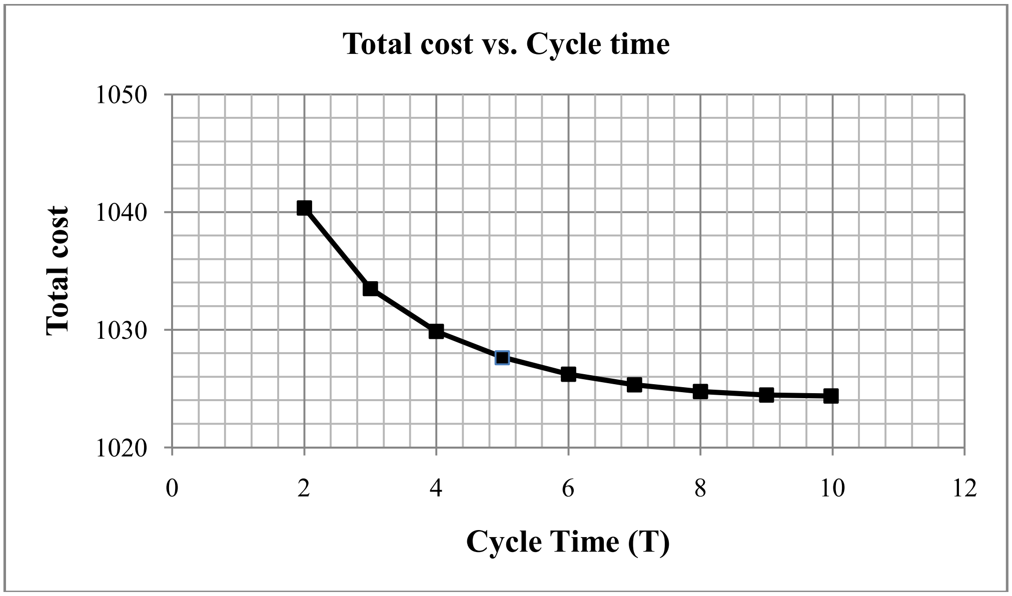

Figure 6.10: Relationship between total cost and cycle time

6.4 Effect of Setup Cost:

In Figure 6.11 and 6.12, the optimal lot size (Q), Cycle time (T) and total cost *(TC)* is potted as a function of unit setup cost *(S).* What is interesting about Figure 6.11 and 6.12 is that the optimal lot sizes increase as the values of setup costs increase, and the optimal costs per cycle increase as the values of unit setup costs increase. This suggests that if the setup costs increases, the manufacturer should produce bigger lot size in order to keep the less number of cycles.

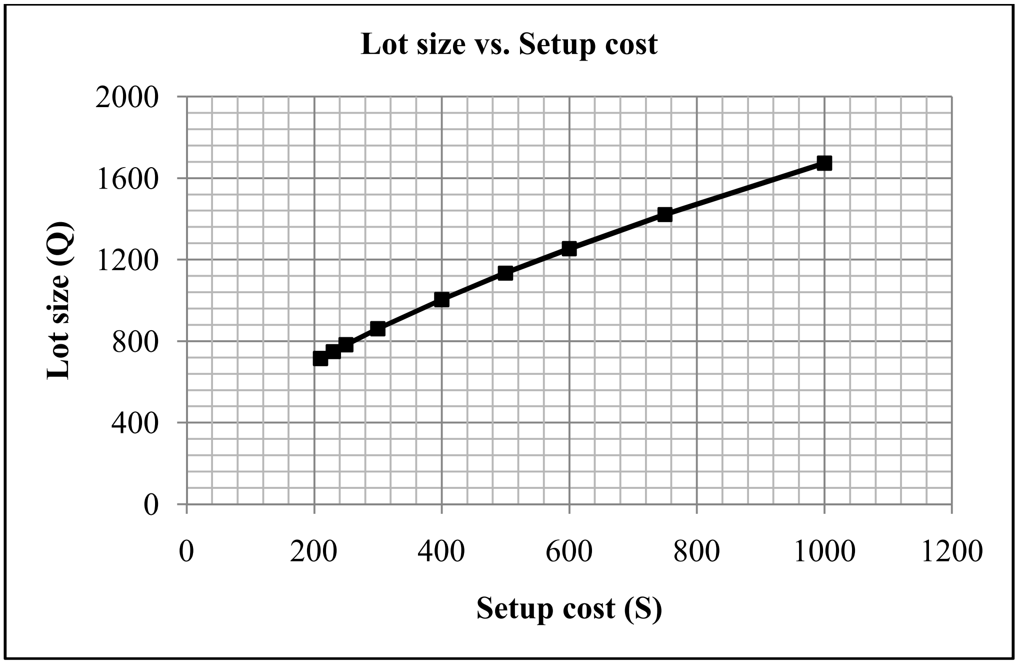

Figure 6.11: Relationship between setup cost and lot size

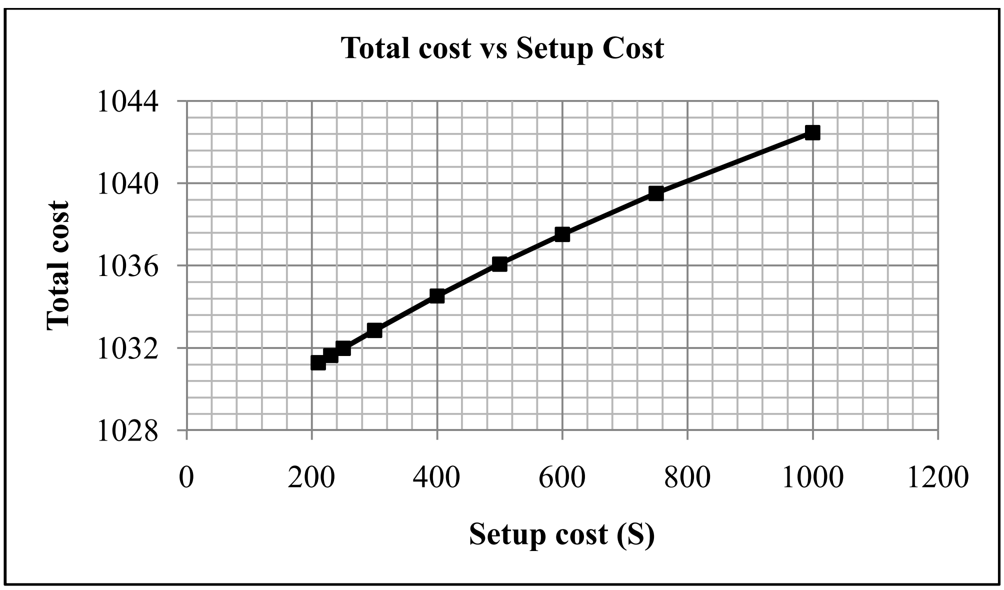

Figure 6.12: Relationship between setup cost and total cost

Cycle time and Setup Cost is also related. With the increase of setup cost, cycle time increases and with the decrease of setup cost cycle time also decreases. The relation is shown in Figure 6.13

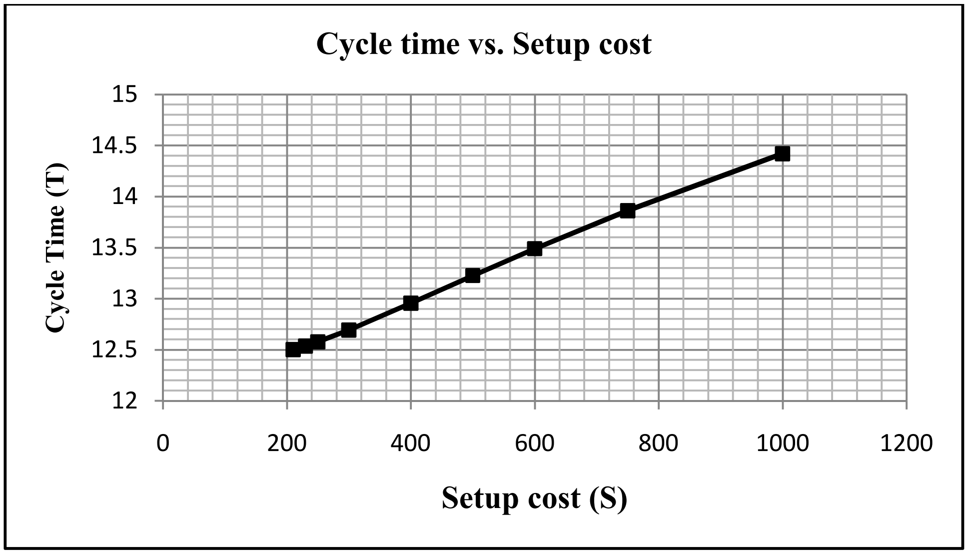

Figure 6.13: Relationship between setup cost and cycle time

6.5 Effect of Holding Cost

In Figure 6.14 and 6.15, we plot the optimal lot size (Q) and total cost *(TC)* as a function of unit holding cost *(h)*. What is interesting about Figure 6.14 and 6.15 is that the optimal lot sizes decrease as the values of unit holding costs increase, and the optimal costs per cycle increase as the values of unit holding costs increase. This suggests that the holding cost increases, the manufacturer should produce less to avoid a big storage cost in the total cost.

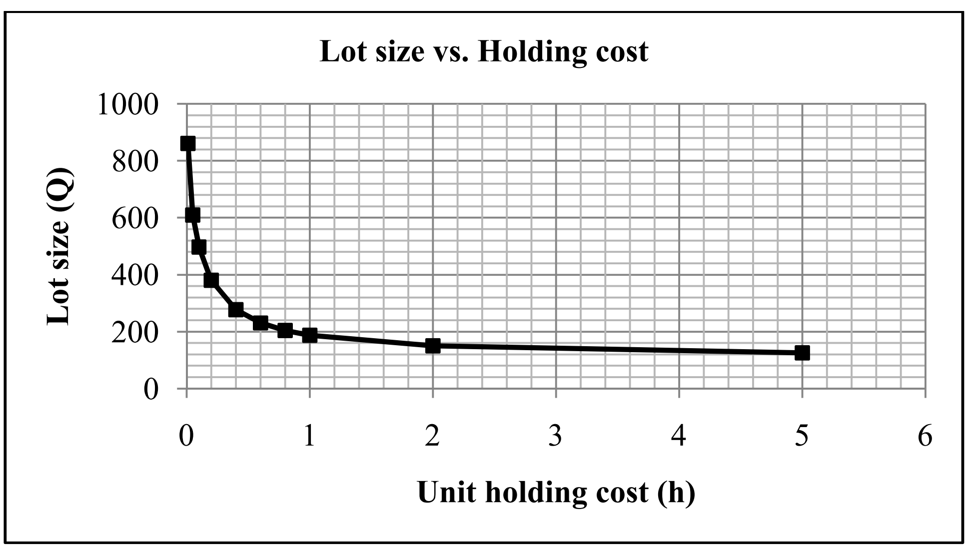

Figure 6.14: Relationship between total lot size and holding cost

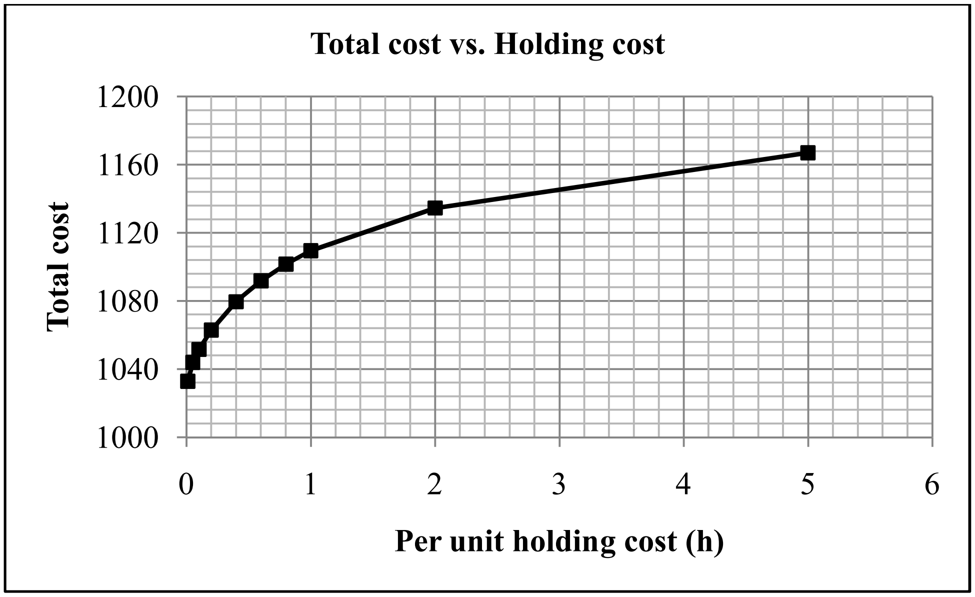

Figure 6.15: Relationship between total cost and per unit holding cost

Cycle time (*T*) and unit holding Cost (*h*) is also related. With the increase of holding cost, cycle time decreases and with the decrease of holding cost cycle time increases. The relation is shown in Figure 6.16

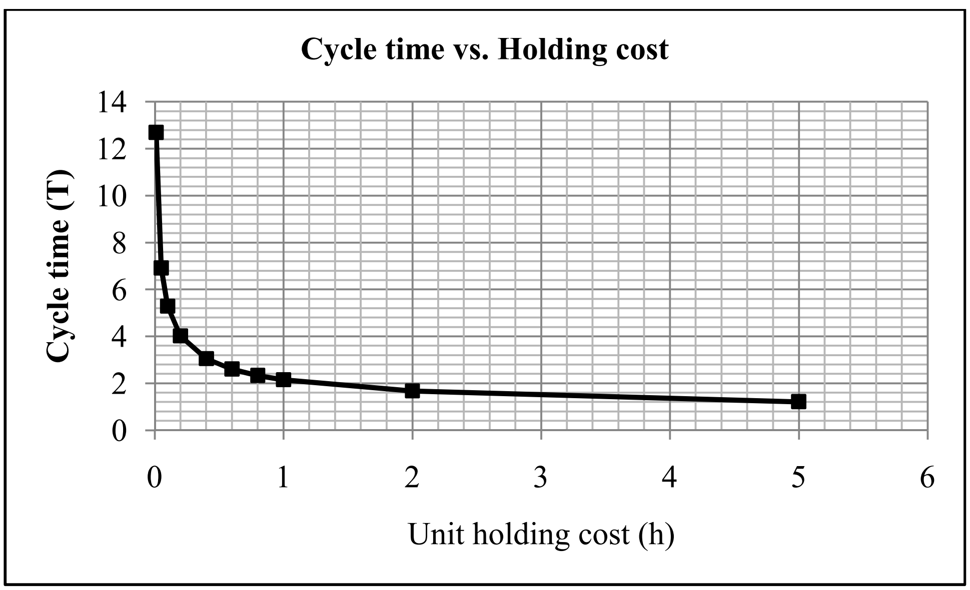

Figure 6.16: Relationship between Cycle time and unit holding cost

CHAPTER 7
CONCLUSIONS AND RECOMMENDATIONS

7.1 Conclusions

Profit of an organization largely depends on its production inventory. Optimization of production inventory model becomes very critical if complex relationship exists between decision variables and objective function. Uncertainty, imprecision and reliability of the production process have significant impact on the production system. It is also important to incorporate uncertainty, imperfection and reliability of the system to optimize the production inventory model.

The objective of this thesis work is to develop a mathematical model of production inventory to investigate the combined effect of production cost, holding cost, setup cost, inspection cost, depreciation and interest cost, defective units cost and backorder cost on cost minimization. Lot size per cycle, reliability of the production process and Cycle time are decision variables. The model is in formulated in unconstrained, non integer non linear form which is complicated in nature and to optimize the model meta-heuristic search algorithm is required. A relatively new technique is PSO which can generate high-quality solutions with shorter calculation time and stable convergence. PSO has been successfully applied to a wide range of applications, but so far gets less attention as a meta-heuristic algorithm to solve such type of complex problem. The evolutionary PSO algorithm is used here to optimize the developed model and the result is compared with GA which is a widely used algorithm. It is found that PSO outperforms GA for the developed model and results less total cost and shorter computational time than GA.

Imprecision and uncertainty in imperfect production process are incorporated in the production inventory problem. Reliability is an important factor for a production process which is incorporated in this model. The model is applicable in an imperfect production process where reliability is an important factor.

7.2 Recommendations

In this thesis reliability is considered as a decision variable which is deterministic in nature and has non integer values. Reliability of the production process depends on a lot of other factors such as production technology, machine capability, work methods, use of on-line monitoring devices, skill level of the operating personnel and inspection, maintenance and replacement policies. In future research, considering reliability probabilistic in nature new EPQ model can be developed. Moreover, demand and chance of defective items can be considered also probabilistic in nature. In this thesis PSO has been used to optimize the model. There are other search algorithms such as ant colony optimization, harmony search algorithm and fuzzy algorithm which can be applied to optimize such type of models. From this thesis it is also found that PSO algorithm has immense potential to be applied in other field of inventory related problem.

REFERENCES

1. Salameh, M. K., and Jaber, M. Y., Economic production quantity model for items with imperfect quality. International Journal of Production Economics, 64, 59–64, 2000.

2. Goyal, S. K., and Cardenas-Barron, L. E., Note on Economic production quantity model for items with imperfect quality a practical approach. International Journal of Production Economics, 77, 85–87, 2002.

3. Teng, J. T. and Chang, C. T., Economic production quantity models for deteriorating items with price- and stock-dependent demand. Computers and Operation research, 32, 297- 308, 2005.

4. Huang, Y. F. and Kuang, H. H., An EOQ model under retailer partial trade credit policy in supply chain. International Journal of Production Economics, 112, 655-664,2005

5. Freimer, M., Thomas, D., and Tyworth, J., The value of setup cost reduction and process improvement for the economic production quantity model with defects. European Journal of Operational Research, 173, 241-251, 2006.

6. Lai, C. S., Huang, Y. F. and Huang, H. F., A deterministic inventory model under quantity depended delay policy using algebraic method. Journal of Applied Science, 6, 406–410, 2006.

7. Leung, K. F., A generalized geometric programming solution to an economic production quantity model with flexibility and reliability considerations. European Journal of Operations Research, 176, 240–251, 2007.

8. Liao, J. J., On an EPQ model for deteriorating items under permissible delay in payments. Applied Mathematical Modeling, 31, 393-403, 2007.

9. Islam, S. and Roy, T. K., Fuzzy multi-item economic production quantity model under space constraint: A geometric programming approach, Applied Mathematics and Computations, 184, 326-335, 2007.

10. Darwish, M. A., EPQ models with varying setup cost. International Journal of Production Economics, 113, 297–306, 2008.

11. Rau, H. and Ouyang, B. C., An optimal batch size for integrated production–inventory policy in a supply chain. European Journal of Operations Research, 185, 619-634, 2008.

12. Panda, D., Kar, S., Maity, K., and Maiti, M., A single period inventory model with imperfect production and stochastic demand under chance and imprecise constraints. European Journal of Operational Research, 188, 121–139, 2008.

13. Hejazi, S. R., Tsou, J. C. and Rasti Barjoki, M., Optimal lot size of EPQ model considering imperfect and defective products. International Journal of Industrial Engineering, 4, 59-68, 2008.

14. Jamal, A. M. M., Sarker, B. R. and Mondal, S., Optimal manufacturing batch size with rework process at a single-stage production system. Journal of Computers & Industrial Engineering, 47, 77-89, 2004.

15. Chen, S. H., and Chang, S. M., Optimization of fuzzy production inventory model with unrepairable defective products. International Journal Production Economics, 113, 887–894, 2008.

16. Chung, K., Hou, K., Lan, S.The optimal production cycle time in an integrated production-inventory model for decaying raw materials. Applied Mathematical Modelling, 33, 1–10, 2009.

17. Huang, Y. F., Optimal retailer's replenishment decisions in the EPQ model under two levels of trade credit policy. European Journal of Operational Research, 176, 1577-1591, 2007.

18. Teng, J. T. and Chang, C. T., Optimal manufacturer's replenishment policies in the EPQ model under two levels of trade credit policy. European Journal of Operational Research, 195, 358-363, 2009.

19. Pal, S., Maiti, M. K. and Maiti, M., An EPQ model with price discounted promotional demand in an imprecise planning horizon via Genetic Algorithm, Computers and Industrial Engineering, 57, 181-187, 2009.

20. Islam, S. and Roy A., fuzzy EPQ model with flexibility and reliability consideration and demand dependent unit production cost under a space constraint: A fuzzy geometric programming approach. Applied Mathematics and Computation, 176, 531–544, 2006.

21. Panda, D., and Maiti, M. Multi-item inventory models with price dependent demand under flexibility and reliability consideration and imprecise space constraint: A geometric programming approach. Mathematical and Computer Modelling, 49, 1733-1749, 2009.

22. Hu, F. and Liu, D., Optimal replenishment policy for the EPQ model with permissible delay in payments and allowable shortages. Applied mathematical modeling, 34, 3108-3117, 2010.

23. Leung, K. N. F., An integrated production-inventory system in a multi-stage multi-firm supply chain.Transportation Research,46,32-48,2010.

24. Ahmed, M.A., Saadany, E.l. and Mohamad Y., A production/remanufacturing inventory model with price and quality dependant return rate. Computers and Industrial Engineering, 58, 352-362, 2010.

25. Sana, S.S., A production–inventory model in an imperfect production process. European Journal of Operational Research, 200, 451-464.

26. Taleizadeh, A. A., Widyadana, G. A., Wee, H. M., and Biaban, J., Multi products single machine economic production quantity model with multiple batch size. International Journal of Industrial Engineering Computations, 2, 213-224, 2011.

27. Wang, X. and Tang, W., Fuzzy EPQ inventory models with backorder. Journal of Systematic Science and Complexity, 22, 313-323, 2009.

28. Seyed, H. R. P. and Seyed, T. A. N., A genetic algorithm approach to optimize a multi-products EPQ model with discrete delivery orders and constrained space. Applied Mathematics and Computation, 195, 506-514,2009.

29. Tripathy, P. K., Tripathy P., Pattanik M., A Fuzzy EOQ Model with Reliability and Demand-Dependent Unit Cost. International Journal of Computational Mathematical Science, 6, 30, 2011.

30. Cheng, T.C.E., EPQ with process reliability and quality assurance considerations. Journal of Operational Research Society, 42, 713-720, 1991.

31. Sadjadi. S. J., Aryanezhad M. B. and Jabbarzadeh A., Optimal marketing and production planning with reliability consideration. African Journal of Business Management, 4(17), 3632-3640, 2010.

32. Tripathy, P. K., Wee, W. M. and Majhi, P.R., An EOQ model with process reliability consideration. Journal of Operational Research Society, 54, 549-554, 2003.

33. Tripathy, P.K. and Pattnaik, M., Optimization in an inventory model with reliability consideration. Applied Mathematical Sciences, 3 (1), 11-25, 2009.

34. Sakaguchi, M., Inventory model for an inventory system with time-varying demand rate. International Journal Production Economics, 122, 269–275, 2009.

35. Yang, H., Teng, J., and Chern, M. An inventory model under inflation for deteriorating items with stock-dependent consumption rate and partial backlogging shortages. International Journal Production Economics, 123, 8–19, 2010.

36. Reza, S. H., Niaki, S. T. A. and Mirhosseyni, S. S., A parameter-tuned genetic algorithm to solve multi-product economic production quantity model with defective items, rework, and constrained space. International Journal of Advanced Manufacturing Technology, 49, 827-837, 2009.

37. Maiti, M. K., Fuzzy inventory model with two warehouses under possibility measure on fuzzy goal. European Journal of Operational Research, 188, 746–774, 2008.

38. Bishop, C. M., Neural Networks for Pattern Recognition. Oxford University Press, chapter 7, pages 253–294, 1995.

39. Kennedy, J. and Eberhart, R. C., Particle Swarm Optimization. In Proceedings of IEEE International Conference on Neural Networks,1942–1948, Perth, Australia, 1995.

40. Kennedy, J., The particle swarm: Social adaption of knowledge. In Proceedings of the International Conference on Evolutionary Computation, 303–308, Indianapolis, USA, 1997.

41. Angeline, J. P., Using selection to improve particle swarm optimization. In Proceedings of IJCNN, 84–89, Washington, USA, 1999.

42. Eberhart, R. C. and Kennedy, J., Swarm Intelligence. Morgan Kaufmann, 2001.

43. Mataric, M. J., Designing and understanding adaptive group behavior. adaptive behavior, 4, 51–80, 1995.

44. Millonas, M. M., Swarms, phase transitions, and collective intelligence. In Artificial Life III, 417–445. Addison-Wesley, Reading, MA, 1994.

45. Salerno, J., Using the particle swarm optimization technique to train a recurrent neural model. In Proceedings of the Ninth IEEE International Conference on Tools with Artificial Intelligence, 45–49, 1997.

46. Bergh, F., Particle swarm weight initialization in multi-layer perception artificial neural networks. In Development and Practice of Artificial Intelligence Techniques, 41–45, Durban, South Africa, 1999.

47. Zhang, C. and Shao, H., An ANN's evolved by a new evolutionary system and its application. In Proceedings of the 39th IEEE Conference on Decision and Control, 3562–3563, 2000.

48. Tandon V., Closing the gap between CAD/CAM and optimized CNC end milling. Master's thesis, Purdue School of Engineering and Technology, Indianapolis, IN, USA, 2000.

49. He, Z., Wei, C., Yang, L., Gao, X., Yao, S., Eberhart, R. C. and Shi, Y., Extracting rules from fuzzy neural network by particle swarm optimization. In Proceedings of the IEEE International Conference of Evolutionary Computation, 74–77, Alaska, USA, 1998.

50. Fukuyama, Y. and Yoshida, H., A particle swarm optimization for reactive power and voltage control in electric power systems. In Proceedings of the IEEE Congress on Evolutionary Computation, 87–93, Seoul, Korea, 2001.

APPENDIX

PSO Code

```
clear all;
close all;
clc;
tic
part_no = 50;
coeff =3;
% tol=10^-25;
a=0.2;
b=1.6;
c=2.8;
n_iter=500;
p_init=rand(part_no,coeff);
v=(rand(part_no,coeff)-0.5)*2;
p_lb=p_init;
p = p_lb;  %correct
y_new=zeros(1,part_no);
y_lb=zeros(1,part_no);
for i =1:part_no
   y_new(i)=shaon(p(i,:));
end
```

```
y_lb=y_new;
z = y_lb';
[val,ind]= min(z, [ ],1);
p_gb = p_lb(ind,:);
%while n_iter < 1000
for n=1:n_iter
    for i=1:part_no
        for j=1:coeff
          v(i,j)=a*v(i,j)+rand(1,1)*b*(p_lb(i,j)-p(i,j))+ rand(1,1)*c*(p_gb(j)-p(i,j));
           p(i,j)=p(i,j)+v(i,j);
           if j==1
               if p(i,j)>=1.0
               v(i,j)=-v(i,j);
               p(i,j)=1.0;
               elseif p(i,j)<=0
               v(i,j)=-v(i,j);
               p(i,j)=0;
               end
           elseif j==2
               if p(i,j)>=10000
               v(i,j)=-v(i,j);
               p(i,j)=10000;
               elseif p(i,j)<=0
```

```
            v(i,j)=-v(i,j);
            p(i,j)=0;
            end
        elseif j==3
            if p(i,j)>=10000
            v(i,j)=-v(i,j);
            p(i,j)=10000;
            elseif p(i,j)<=0
            v(i,j)=-v(i,j);
            p(i,j)=0;
            end
        end
    end

    y_new(i) = shaon(p(i,:));
end

for k=1:part_no
    if y_new(k)<y_lb(k)
        p_lb(k,:)=p(k,:);
    end
end
for m=1:part_no
```

```
    y_lb(m)=shaon(p_lb(m,:));
    end
    z = y_lb';
    [val,ind]= min(z, [ ],1);
    p_gb = p_lb(ind,:);
    resultn(n)=y_lb(ind);
end
[p_gb y_lb(ind)]
plot(resultn)


 function ynew = shaon(p)
C=1;    % (1-2500) unit production cost
D=1000; % (50-100,) unit per day
P=1500; % unit per day (must be greater than demand)
h=.01; % (0.5-0.01) per unit per day
s=100; % (100-1500, 50-5000) per cycle
l=1500; % constant (800-1600) constant
m=0.5; % constant (0.2-0.8)
n=0.75; % constant (0.5-0.9)
I=0.05; % (1%-20%)
J=0.05; %(1%-20%)
A=0.1; % (backorder administrative cost)
V=0.1;  % (backorder cost due to loss of goodwill)
```

```
t=h+V;

R=A*D*h*t^(-1)-A*D*h*V*(1/2)*t^(-2);

ynew = R+C*D*p(1)^(-1)+(h/2)*p(3)-(D/2)*(h/P)*p(3)*p(1)^(-1)+ s/p(2)+l*s^(-m)*p(1)^n*p(2)^(-1)+I*D*p(1)^(-1)+J*p(2)^(-1)*p(3)-J*p(2)^(-1)*p(3)*p(1)+p(3)*h^(2)*V*(1/2)*t^2+D^2*A^2*p(3)^(-1)*V*(1/2)*t^(-2)-D^2*A^2*p(3)^(-1)*t^(-1);
```