Supplier Selection Problem: An Approach Using Genetic Algorithms

Arash Ashkan Alam

A Thesis

in

The Department

of

Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science (Industrial Engineering) at

Concordia University

Montreal, Quebec, Canada

July 2010

Your file Votre référence
ISBN: 978-0-494-70968-9
Our file Notre référence
ISBN: 978-0-494-70968-9

ABSTRACT

Supplier Selection Problem: An approach Using Genetic Algorithms

Arash Ashkan Alam, M.A.Sc,

Concordia University, 2010

This research tackles a supplier selection problem composed of different suppliers with limited capacities, a client with deterministic multi-period demands and specific allowed inventory limit in each period for a single product. The objective is to select the most economical set of suppliers in order to meet the client's demand. A novel genetic algorithm and chromosome representation are proposed to find near optimal solutions. The performance of the proposed algorithm is compared with the exact approach using randomly generated data sets. In this supplier selection problem, initially proper population size is determined for three different problem sizes of small, medium and large; as the next step of the experiments, proper numbers of iterations for each problem size are found; finally, different mutation probabilities are tested for different problem sizes and the best mutation probabilities for each problem size are selected based on the calculated error. By the help of the results of the experiments and gathered information, proper population size, number of iterations, and mutation probabilities are recommended for problems with similar size and constraints.

Key words: Supply Chain Management, Supplier Selection, Genetic algorithms, Optimization

Acknowledgements

I would like to express my gratitude to my supervisors, Dr. Bhuiyan and Dr. Chauhan, for their support and guidance in my thesis. Their encouragement and useful comments have enabled me to reach this education level.

I would like to appreciate Dr. M. Defersha, my friends, and my colleagues for their precious comments and critiques in fulfillment of my thesis.

I profoundly thank my parents and my sister. Without their love and encouragement, this work would not have been possible.

Table of Contents

List of Figures

List of Tables

1. Introduction

A supply chain is a network which connects customers, suppliers, inventories, and retailers, and each role plays part in providing customers with a service or product. The management of this chain includes the process of dealing with demand and managing the flow of products, work in progress, or raw materials. One of the most vital problems in supply chain management is supplier selection, where the firm or the customer needs to make a decision towards ordering its demands from different suppliers while meeting all requirements and budget limitations; the firm tries to select one or many suppliers. Knowing all the necessary information and data about the demand will lead to meeting the requirements.

Companies have different procedures for supplier selection. They may have different stages such as pre-selection, selection, and post-selection stages; in each stage, based on available data, which includes demand and historical experience, with other members of the supply chain, decisions regarding the amount to purchase, time intervals, and set of suppliers are determined. Another key factor in supplier selection is the delivery method. Moreover, different suppliers have different capacities. Firms usually have short-term and long-term plans for supplier selection. For instance, firms have clear policies towards replacing a supplier due to different reasons, such as late deliveries or poor quality. Another factor in supplier selection is pricing. One of the goals of the supplier selection procedure is reducing and optimizing the costs. Shipping cost, lead time, quality, and stochastic or deterministic demand are other factors. Suppliers may need to meet demand for a single or multiple periods. In order to have an efficient supply chain, a suitable supplier selection procedure should be undertaken. In this research, the supplier selection problem is composed of different suppliers with given capacities for a single product. The client has a deterministic demand and is allowed to hold a specific level of inventory at the end of each period. In order to solve this problem, a genetic algorithm (GA) is proposed. GA is a heuristic search technique; the idea of this method is taken from the evolutionary ways of natural selection and genetic science. In this research, GA is utilized to find the optimal solution for the proposed model. In GA, an initial population exists which represents a set of feasible or infeasible solutions to the problem. In different steps and through the help of genetic operations, crossover and mutation, this set of solutions is improved.

This thesis is organized as follows. Chapter 2 presents a literature review on the topics of supply chain management, supplier selection, optimization, and genetic algorithms. Chapter 3 presents the problem statement and solution approach. Chapter 4 discusses computational results. Chapter 5 presents future studies, and finally Chapter 6 provides conclusions to the research.

2. Literature review

The focus of this thesis is supplier selection. In this section, initially a review on supply chain management is presented. Since supplier selection problems are an important part in the field of supply chain management, the literature on supplier selection then will be reviewed. A brief introduction on optimization techniques will then be given since optimization plays an important role in this work. Finally, a review on genetic algorithms and other metaheuristics is conducted. The GA is the utilized search technique in this research.

2.a Supply Chain Management (SCM)

Supply chain management (SCM) problems have always been a point of interest for researchers. SCM covers a vast range of problems including inventory management, optimization, scheduling, planning, and transportation. Researchers and scientists define supply chain and supply chain management in different ways. Tan et al. (1998) defined a supply chain as a network which includes management of supply for services, materials, or products. This supply of products includes the raw materials from the very beginning to the end customer and even the process of reverse logistics and recycling them for future demand in the same industry or different ones. The goal of this approach is to help firms to increase their benefit and getting advantage of different members in the supply chain. Tan in his definition believed in expanding the intra-enterprise activities and making stronger connections between partners.

Saunders (1995) defined the supply chain as the whole chain of trade between different parts including the initial source of raw material, different firms involved in adding value to raw materials and work in progress and finally retailing to the end customers. In this definition, he mostly concentrated on the firms, their role in the chain, and the relationship between all the members of the supply chain.

Scott and Westbrook (1991) defined the supply chain management as a network which connects all the members engaged in different activities related to procurement of raw materials, making products, and retailing them to the end customer. Each member of this network has its own limitations defined by different constraints in the network or real world constraints.

Saad et al (2002) believe supply chain management is an innovative phenomenon in recent decades. They believe:

- Supply chain management is a process not a single task and there are different elements which have different effects on it.
- It is not a short process; it is a process which needs time and it should be amended and enhanced over the time to get the best possible results.
- In the supply chain, different levels are available; in order to have it more effective, research and development is needed to generate new ideas, increase the knowledge level, and come up with more alternative solutions during the time.

These will help in planning and defining more realistic goals and achieving them will be easier.

- The relationship and communication between different parts of the supply chain including the parts inside each firm will be managed and the flow of information will be easier.
- It is not a short term process; it needs a lot of research and continuous study besides long term and strategic planning.
- In order to have a successful supply chain policy, higher management commitment is necessary.

Among all the mentioned reviews, there are common elements which are the network, material, and customer.

2.b Supplier Selection

The main concentration of this thesis is supplier selection. In this part, the general concepts of supplier selections will be initially discussed; afterwards, existing research in this area will be reviewed. The main interest of this research is in tactical level supply chain planning where the demand in each period is known with high certainty. However, various researchers proposed several variants of supplier selection problems in the literature. For instance, Burke et al. (2007) studied the procurement of a particular product for one period with stochastic demand. They believe that in order to have successful supply chain management, an effective sourcing strategy should be found and implemented. They simultaneously studied product prices, supplier cost and capacities, historical supplier credibility, and firm specific inventory costs. They also implemented a specific supplier diversification function of a firm. In their numerical analysis, they defined different parameter ranges and studied its effect on different factors including firm profit, optimal number of suppliers, and quantities. In addition, they also studied the effect of minimum order quantity on the sourcing strategy, and the relationship between minimum order quantities and reliabilities. Their research showed that procurement of one product is a principal strategy in case supplier capacities are large relative to the customer demand; moreover, the customer should not get benefits due to the variation of products.

Later, they studied the effect of variations in supplier pricing policies and the restrictions on suppliers' capacities towards the optimal sourcing policy for a single customer. In their paper, analytical and numerical analyses are conducted (2008). They used three different types of discount policies and developed three heuristic algorithms to address the problem. In their model, they found at most only one of the suppliers has to supply less than its specific capacity and they also validated their claim towards effectiveness of their heuristic model for multiple supplier selection problems. They believe their method leads to a near optimal solution for similar problems.

Liao and Rittscher (2007) improved a multi objective programming model, where they faced a typical non-linear mixed integer combinatorial optimization problem. Their problem was about purchasing a single product in different periods with given demands.

The authors also mixed various factors including selecting the suppliers, lot sizing, and transportation decisions. They designed and implemented a novel GA with problem specific operators. The goal of the proposed genetic Algorithm was minimizing the logistic cost. This cost was incurred due to the purchasing and ordering cost, the inventory holding cost and the transportation cost; moreover, the mentioned cost also includes the cost incurred due to quality tests and rejections and also expenses incurred due to the late deliveries. They believed their proposed GA successfully addressed the problem and it could also address similar problems.

Later, Liao and Rittscher (2007) studied a supplier selection problem which had to meet different objectives with demand uncertainties. Their model faced similar constraints like their previous research and had to minimize the total cost. This total cost, the same as before, was incurred due to purchase cost, quality tests and rejection, and late delivery cost. Moreover, they extended the supplier flexibility as well as demand and timing uncertainties in their new research. In order to solve this problem which has probabilistic demand, a multi-objective model was implemented which also meets the demand uncertainties. They believe that in order to specify the supplier selection under stochastic conditions, some specific factors should be considered at the same time; the most important factors in this problem are total cost, quality rejection rate, delivery delay rate, and flexibility rate. As in their previous paper, they tackled this problem using a GA to study their non-linear mixed integer combinatorial optimization problem. They found that the quantity and timing of uncertainties are the most probable deviations in any supply chain. These uncertainties could always cause both customers and merchants different problems and dissatisfaction.

Weber and Current (1993) designed and implemented a multi-objective approach in their paper for a deterministic supplier selection model. They believed that the procurement of the demand from different vendors in each supply chain leads to significant costs and problems for different firms engaged in the supply chain. These costs and inconveniences are caused by different factors such as suppliers' lack of flexibility, uncertainty in demands (for a single or different periods), or even not fully meeting demand. They validated their arguments through the help of numerical experiments. The GA they used was applied to a deterministic model.

Qi (2007) combined two often separately studied issues in a case in which suppliers have different capacities and the demand quantity is based on the presented price. He designed and implemented a heuristic algorithm and an optimal dynamic programming algorithm to determine the optimal selling price while suppliers have enough capacity to supply demand. Through the help of these derived quantities, the goal of maximizing the total profit can be achieved. Finally, he validated the results by numerical and computational experiments to determine the effectiveness and efficiency of the proposed method. This paper also provides two strategies for managers: in the case that sourcing information is not at hand, it is superior to have a conservative production plan. In case that the market demand information is unidentified, it is superior to make an aggressive production plan.

Kheljani et al (2009) considered suppliers and customers in the logistics; they agreed that solely concentrating on one party in the logistics does not logically lead to optimizing the benefit for the whole logistic or the supply chain. In their research, they took into consideration a supplier selection problem; this problem was assumed in a centralized supply chain. They studied co-ordination among one customer and multiple potential suppliers which leads to choosing the supplier. By their global view, they not only minimized the total cost of the supply chain, but also increased benefits of all members present in the Supply Chain. They used mixed-integer non linear programming in order to solve their proposed model. In their numerical exam, they tested two different cases of purchasing a single product from three different suppliers. In Figure 1, a centralized supply chain (single buyer and multiple suppliers) is shown.

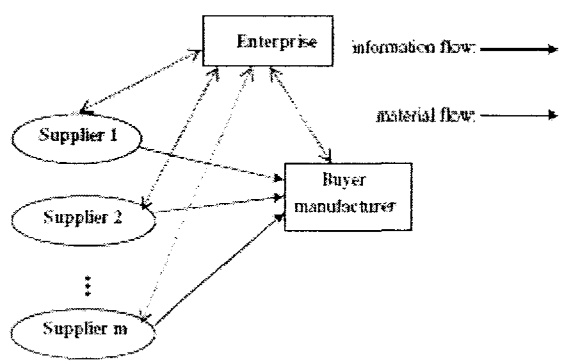


Figure 1 - Centralized supply chain (single buyer and multiple suppliers), (Kheljani et al, 2009).

Through the help of the Lagrangian relaxation, Benton (1991) studied a manager's task towards purchasing resources with constrained order quantity. In his research, alternative pricing schedules from multiple suppliers were available. The most important constraints in this research are on budget and allowed inventory level. These constraints are for up to ten items offered by three different suppliers. The discount model imposed on the problem was all units discount in three different types. This discount model is applicable for each item separately. The goal of this model is optimizing the total cost and total inventory holding cost. The manager is responsible for selecting only one supplier. In case of having multiple suppliers, the optimal solution is 8% less than that of the single supplier.

Chauhan and Proth (2003) studied a problem in which a single product with fixed quantity is attempting to be purchased from multiple suppliers. In their model, each firm has its own pricing system. In addition, all the suppliers in this model have their own specific capacity and also a minimum allowed purchased amount; they believe a certain provider may be the most inexpensive for more than one manufacturing unit; thus choosing this provider for one of the manufacturing units may make other ones select providers with higher prices for their services. They later expanded their model and studied for different suppliers and different customers in the supply chain.

Chopra and Meindel (2004) listed four key source-related processes, as shown in Figure 2.

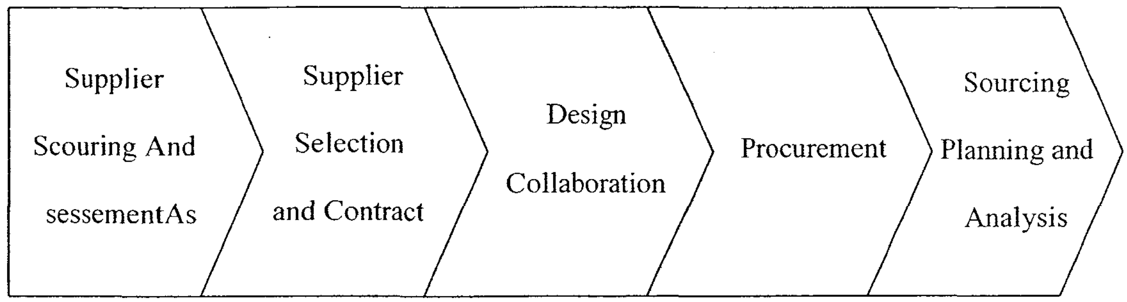


Figure 2 - Key Sourcing-Related Processes, (Chopra and Meindel, 2004)

Like many other researchers, they also had the idea of studying the supply chain as a whole and not only a single part of it. They believe effective sourcing strategies in any member in the supply chain can increase benefits not only for a single part of the supply chain, but also for other members and consequently for whole the supply chain.

Rosenblatt et al (1998) studied another supplier selection problem. As most in supplier selection problems, a series of different potential suppliers which could meet the requirements are given. The objective is to determine from which supplier, at what amount and how often purchases should be made. They presented all the requirements for the optimal solution in their paper and made a connection between the presented method and a method which uses a single source as it is proposed by Just In Time (JIT) concepts. As it was mentioned, the problem should be solved for each period; in each period, the amount of supply from a specific supplier should be determined. Moreover, it was the task of researchers to determine the length of each period. They presented the fact that by using their model, the researchers are able to decrease errors as much as they want.

Chang (2006) presented a novel supply strategy to deal with a problem with one product and many suppliers. The problem he worked on had limitations which were inspired from a real industrial case and used the price-quantity discount (PQD) policy. His paper presented a set of linearization strategies which could easily be coded by a programming language to determine the best procurement policy while reducing the inventory holding cost. He believed his proposed method is efficient in real-world problems.

Ho et al. (2010) conducted comprehensive research towards different techniques which were utilized in order to address different supplier selection problems. They highlighted the fact that supplier selection and evaluation problems have been a great source of interest for researchers. Based on their conclusion, only 1.28 % of researchers took advantage of pure GA's to tackle their understudy problems. In addition, they highlighted the fact that many researchers combined GA's with other techniques such as multi-objective programming, fuzzy, and artificial neural networks (ANN).

Keskin et al. (2010) studied a supplier selection problem with the help of Fuzzy Adaptive Resonance Theory (ART) algorithm. They believed the process of supplier selection and supplier evolution is an elaborate procedure which is affected by different decision factors. They also highlighted the fact that a huge number of techniques exist and have been tried in order to address supplier selection problems. They named also mathematical programming techniques as an important technique to address these problems. In their research, multiple products from different suppliers based on discretion of decision making committees should be purchased and proper suppliers should be evaluated and selected. By using multiple desired factors and by the help of proposed Fuzzy ART selection algorithm, the final result which is selecting proper suppliers can be determined. This brand-new research mostly concentrates on qualitative aspects of the supplier selection procedure. The effect of multiple periods is not studied here. Finally, the minimum capacities of suppliers are not taken into consideration.

Sawik (2010) studied the optimal demand allocation to a set of qualified suppliers and modeled it as mixed integer programming. In his research, the suppliers work under make-to-order production conditions. He initially studied the problem with a single objective under the condition of existence or lack of discount policies. He then studied the problem with multiple objectives. In this paper, the factors utilized towards supplier selection are price and quality of the products. In addition, he considered the reliability of suppliers in meeting the delivery deadlines as another important factor towards supplier selection. As an important difference of this paper with other papers in field of supplier selection, the supply chain has a single producer which performs the assembly task by utilizing products procured in the previous levels of supply chain. Although in this paper suppliers have different capacities, they are not restricted to a minimum obligatory delivery limit. The prices and the quality of different suppliers are not the same. Besides, this paper only addresses the supplier selection problem for a single period. Finally in their model, the expected defect and late delivery rates are taken into consideration.

Ebrahim et al. (2009) implemented a scatter search algorithm in order to address a supplier selection problem with different discount schemes. In their model, they imposed qualitative factors beside quantitative factors. In addition, they simultaneously took three different discount policies into consideration. They brought these discount policies as an additional objective function to their model. The models utilized in their paper are all-unit discounts, incremental discounts and finally total business volume discounts. In their problem, the selection procedure is conducted for a single product in a single period. They argued their problem is NP-hard; based on this fact, they designed and implemented a scatter search algorithm (SSA) in order to address this complex problem. In their approach, they tried to minimize the total purchasing cost with existence of discount schemes; moreover, they also tried to reduce the total number of defective items up to minimum possible amount. As the final objective, they tried to minimize the total number of late delivered merchandises. They compared the results of their algorithm with LINGO package solutions for 24 sample problems; as a result, the comparison showed the algorithm's ability to find high quality solutions in a short period of time. In this paper, the authors mostly concentrated on different discount schemes and applied it only for a single product in a single period. In addition, solving the problem for a single period prevents the deals on inventory holding policies caused due to existence of different periods.

From the preceding review, different cases of supplier selection problems were discussed. In one case, a single product for a single period was studied. In some cases, the customer was dealing with capacitated suppliers. In other cases, backorders were not allowed. Besides, different search techniques were used to address different problems including Lagrangian relaxation, dynamic programming, and GA. In some cases different discount policies were also imposed. In addition as it was seen here, most of the reviewed recent papers concentrate on different discount policies and do not consider the suppliers' restrictions. Moreover, these problems were mostly solved for a single period which is not the same as the present work. The gap in all of these research papers is addressing a problem with different capacitated suppliers, for different periods through the use of GAs while holding inventory is permitted. In some problems due to the increasing complexity, linearization techniques were used and as one of the strength of the proposed technique, it is not necessary to do the linearization for many of the constraints.

2.c Optimization

Optimization techniques are methods which are used to find optimum solutions for decision problems. Different models require different techniques. Biegler and Grossmann (2004) categorized different types of optimization problems. They believed optimization has great effects not only on academic areas, but also on industrial fields of work and study. In their research, they mainly concentrated on process systems engineering. They initially categorized optimization problems into two fields with continues variables and problems with discrete variables.

Sahinidis (2004) studied different types of optimization problems under uncertainty and their related complexities. He discussed different types of programming including stochastic programming, fuzzy mathematical programming, and stochastic dynamic programming. Then he discussed different applications of each type. For example for stochastic programming, he hinted to its function in the field of agriculture, aerospace industry, and sales planning. For fuzzy programming, he named various fields of usage including production planning and the transportation problem. Finally for stochastic dynamic programming, he mainly tackled multi stage decision making problems. In addition, similar to fuzzy programming, he mentioned different usages for stochastic dynamic programming including production planning or aerospace industry. For each type of mentioned problem, the author presented and referenced different types of algorithms and he believed many of the algorithms had been successfully used in different real world problems.

Hillier and Lieberman (2005) presented mentioned algorithms like simplex which are able to lead the researchers to the optimal solution do not always effective for all problems with different size or with different complexity. These algorithms are used to solve different models such as linear and integer programming. In some cases, these algorithms are not able to solve the problem due to the problems complexity; in this case, researchers usually accept feasible near optimal solutions as well; the methods which likely lead researcher to these near-optimal or optimal solutions are called heuristic methods. There is no guaranty for these techniques to find the optimal solutions, but a well-designed heuristic could help the researcher to reach a near-optimal or optimal solution. The other problem with heuristics is the fact that each heuristic is designed for a specific problem and does not certainly work for other problems; due to this restriction, a more power technique was developed and called metaheuristic. A metaheuristic is a general technique to find a heuristic solution for a specific problem.

Genetic Algorithm (GA) is a type of metaheuristics which is different from other types of metaheuristics. Tabu search and simulated annealing are also two well-known metaheuristics. Hillier and Lieberman (2005) presented tabu search, simulated annealing, and GAs in their book. Tabu search is a technique which is based on a hill climbing strategy and there is always a risk of cycling back to a previous local optimum which should be prevented by forbidding specific moves.. Moreover, simulated annealing is constructed based on finding the tallest hills strategy and this issue needs enough time. This approach usually starts with a feasible solution and next step is taken based on this solution although it is difficult in many cases to efficiently find this initial solution. The main difference between the two previously discussed methods and GAs is the number of possible solutions analyzed together. GAs work on all available chromosomes of the current population at the same time. More details about the way GAs work will be explained in the following chapters.

As previously mentioned, optimization techniques should be used to find a solution for mathematical models which could be solved. In this research, linear optimization techniques and GA are simultaneously used to find the best possible answer for the presented model.

2.d Genetic Algorithms

In this part of the research, the most common heuristic methods will be initially introduced and discussed. In the next step, the advantage of the method implemented in this research will be presented.

Three different metaheuristic techniques are usually preferred by the researchers to address NP-hard problems. These techniques are genetic algorithms (GA's), tabu search (TS), and simulated annealing (SA).

Kita and Tanie (1997) introduced genetic algorithms as an optimization method which is directly inspired from the genetics and biological sciences. In genetics, a great part is allocated to evolutionary studies. The authors believed this method was inspired from animals and plant evolution. As in any optimization problem, a search space should be defined and navigated in each step. From their point of view, each possible solution or proposed answer could be presented as a chromosome as in genetic science. These chromosomes are amended at each iteration by the help of crossover or mutation operations.

Youssef et al (2001) conducted a comparative study of GA, TS, and SA. They named these methods as general techniques with an iterative nature for addressing combinatorial optimization problems. Although these algorithms have many similarities, they have many different specifications, specifically in their strategies in exploring the search space. The most important similarities are:

1) These techniques are just approximation methods and they do not assure whether the final result is global optimum or not.
2) The termination condition should be defined by the researcher.
3) These methods are general techniques and could be adjusted and specialized for different problems.

Two important factors exist which help the researchers towards favoring one technique to other ones. These two factors are the required time for running the algorithm and the quality of the solution which is defined as the deviation from the optimal solution. Some other factors also exist which are as follows:

1) The solution encoding: in each technique, the possible candidate solutions should be transformed to an acceptable format for the algorithm which could be easy or difficult.
2) The initial set of candidates or solutions as the starting point of the algorithm and its generation technique: all the mentioned search techniques should start from a point in the search space. These candidates should be generated either randomly or based on a specific set of rules regarding the type of the algorithm.
3) Evaluation technique: At the end of each iteration, the current set of solutions should be evaluated and ranked based on a predefined set of rules. In addition by the help of this technique, acceptable and unacceptable solutions will be determined.
4) Algorithm operators: through the help of these operators, a new set of solutions or candidates for the next iteration will be generated or derived from the current existing set.
5) Parameter assignment: this technique determines the value of different parameters at each iteration.
6) Termination condition: this condition determines when the algorithm should be stopped.

The authors introduced SA as an iterative search technique which is designed based on the metals annealing procedure. Similar to GA's, an initial set of solutions or candidates should be in hand. The set should have variety. In this approach, a temperature function, T, is designated to determine the hill climbing policies for each candidate at each iteration. Another specification of this approach is performing the partial search of the search space. By decreasing the value of the T function, the uphill moves will have less chance to be done. On the other hand based on this fact, the search will eminently be more random while at low temperatures it works more greedily.

The authors also discussed TS in their paper. In TS, a set of feasible solutions are in hand and the algorithm is trying to amend and improve the current solutions. In this technique, the algorithm selects different directions to perform several moves. As the next step, this algorithm selects the most efficient move which leads to the best answer. These best answers are selected and gathered. The size of candidate solutions is determined by comparing the quality and performance. At each step, chances of reverse moves always exist. This issue is addressed by defining tabu moves. These tabu moves make the algorithm able to escape from local optimums. The list of tabu moves should be defined for the algorithm and at the end of each iteration, and the features of new moves should be memorized by the algorithm. During different iterations, the tabu list maybe changed.

SA requires adequate variety and evaluation functions which make this algorithm more complicated to be implemented for the problem under study in this research. One of the key factors in this method is T function; this function value determines whether the candidate in hand could be the next under-study solution or not; in addition, it determines whether this candidate is not already an evaluated solution among current solutions in hand (Hillier and Lieberman (2005)). The T function also causes more complexity for this problem while it could be addressed more easily through the help of GA. Finally, the core procedure of the SA algorithm directly works based on the T function; in case of any mistake in the T function, the core procedure would probably be severely affected. In addition as the first problem of TS, this technique usually stars from a set of feasible candidates which is not the same as GA.

Arostegui et al (2006) also did a brief comparison between GA, SA, and TS. They discussed a class of problems which are very difficult to solve optimally and these problems are classified as NP-hard. Although some algorithms to solve these types of problems optimally are in hand and used for several small cases, these algorithms suffer from a huge amount of calculations and combinations which make them almost far from successful implementation since they need a huge amount of time resources. They proposed GA, SA, and TS as the most popular general heuristic techniques. The authors highlighted the lack of various in depth comparison studies between GA, SA, and TS. They highlighted the fact that different problem domains may be solved more efficiently by one of the mentioned algorithms while for some domains the mentioned algorithms may lead to very close results; this fact needs to be proved by the help of experimentations in different domains. In their research for a facility location problem, they concluded TS had a better performance in comparison with SA and GA; they proposed TS as the initial approach for the similar problems.

Yun et al (2009) developed a hybrid GA with adaptive local search technique to deal with multi-stage supply chains. In their research, they faced a complex problem and they believed counting all the possible feasible solutions would not efficiently work. In their model, they faced a complex structure which could not easily be solved by the help of conventional algorithms. They also mentioned that the chance of not finding an answer for the model by the help of conventional algorithms also exists. They believed in GA's as alternative techniques which is able to overcome the weakness of conventional algorithms. They also highlighted the proven theory that GA's are more efficient in finding the optimal or near optimal solution of complex supply chain problems instead of conventional algorithms. Furthermore, GA's are sometimes inefficient towards finding optimal solutions for problems with complex values; they address this issue by the help of hybrid GA (hGA) with conventional local search technique. As a general fact, it is mostly known that hGA's have better performance in comparison with GA's.

Yimer and Demirli (2010) used GA to solve a two-phase optimization in a scheduling of a dynamic supply chain. The authors mentioned their problem has a complex search space with many solutions. Due to the complexity and demand for a fast exploration in the search space, they used GA. They also named GA's in their paper as an efficient search technique for problems with large search space. In addition as an important factor towards favoring GA to other heuristic methods, GA is usually able to scale and explore the search space with less amount of information in comparison with SA and TS. These required pieces of information could be problem convexity or objective function differentiability. They also hinted the fact that GA's are usually used to find the optimal or near optimal solutions. They also similarly solved a mixed integer linear programming (MILP) by the help of their proposed GA and they found their proposed technique efficient and satisfactory.

Reid (1996) studied a constrained optimization problem through the help of GA. He believed GAs could be used for solving different types of optimization problems. He indicated that these algorithms are a part of probabilistic relaxation methods. He studied different problems with linear equality and inequality constraints. The objective function could be linear or nonlinear. In his research, he presented new methods including the two point crossover operator, half feasible crossovers, and feasible mutation. This method may impose extra unnecessary limitations for exploring the research space.

Xing et al (2006) developed a novel GA named Intelligent Genetic Algorithm (IGA). IGA is used to find the global optimum for different optimization problems; these problems should be in the classification of multi-minima functions. In their paper, they discuss the required factors of an IGA. In the next step, they explained the way the intelligent genetic algorithms evolve for tackling the specific optimization problem by the help of different genetic operators. At the end, the result of the previous step is tested through numerical experiments. The results of these tests were compared with the existing global answers; they believed their approach is more efficient than other algorithms.

Borisovsky et al (2007) studied two different genetic algorithms for a supplier selection problem. In their problem, they faced a single product purchase in a single period from different suppliers. Suppliers in this problem have their own specific capacity. They are also restricted to a specific minimum amount of supply for each supplier in case they are selected. They used two different types of GAs in their study. Their first algorithm was with binary representation while the second one was with non-binary representation. In their numerical experiment, they tried a different number of suppliers and customers and different ordering fixed cost. From the results, they concluded that a new and efficient genetic operator could be reached by the help of combining the branch-and-cut method for solving mixed integer programming problems and the recombination techniques.GA is a search technique which is used in this research.

Based on information presented here and by reviewing different research works a tendency for selecting GA for different supplier selection exists and the results were satisfying from the authors' point of view. Different supplier selection problems had different conditions, requirements and constraints which are different from the one studied in this thesis. Based on these facts, GA is the favored approach to solve this problem. Through the help of GAs, as a strong search space browsing technique, this problem will be studied.

In the next chapter, the problem statement and approach will be discussed.

3. Problem statement and solution approach

In this section, initially the problem and the formulation are presented. In the next part, a GA based solution approach is proposed to address the problem. Finally, the algorithm is discussed.

3.a Problem Introduction

In this research, a supplier selection problem is studied where a set of suppliers have different capacities; each suppliers' capacity remains constant during all the periods. All the selected suppliers should supply to the client a known agreed quantity all along the planning horizon. An important limitation in this problem which makes it different from the typical supplier selection problem is the existence of a minimum amount of supply which means there is a minimum amount that each supplier should supply if it is selected. It is assumed that the demand for each period is known. The customer is allowed to hold inventory for its prospective demand but backorders are not permitted. For simplicity it is assumed the initial inventory level at the beginning of the planning horizon is zero. The objective of the problem is to select a specific set of suppliers at the beginning for all periods to minimize the total cost while demands are met; the total cost is incurred due to the purchasing of the product and holding the inventory. The payments for the purchased product are instantly made and the effect of delay in payments is not taken into consideration in this problem. It should be noted that the selected supplier set is valid throughout the planning horizon. At the end of each period, the remaining inventory level should not be more than a given limit set for that period. Borisovsky et al (2009) studied a supplier selection problem in which a set of suppliers provide a single product for a set of customers during a single period. In their research, the suppliers are also similarly capacitated like the problem presented here. While minimizing the total cost, the supplier selection should also be conducted. Moreover, they imposed a fixed ordering cost due to selecting the specific supplier which is eliminated here. The major difference added here is the effect of holding the inventory and carrying it forward to be utilized in the next period. These added constraints are very common in real world industrial problems which are neglected in their paper.

The notations used to represent the data and decision variables are shown in Table 1. This table shows a matrix in which suppliers are at the left side of each row and periods are at the top of each column. X_{ij}, the meeting point of row "i" and column "j" shows the amount of purchase from the supplier i in period j. Suppliers vary from i to m, in different period j to n. Two rows at the bottom of the table show the allowed inventory level for each period I_j and demand D_j for each period.

The demand in each period is met through the help of the purchase in that period plus the inventory from the previous period; there might be some amount left at the end of each specific period as the inventory of that period. As previously mentioned, inventory at the end of each period is shown by I_j and it is the difference between the demand and the sum of the previous period inventory and current period demand.

Table 1 – Suppliers, Periods, Allowed Inventory Levels, and Demands

		Period					
		1	2	3	4	...	n
Supplier	1	X_{11}	X_{12}	X_{13}	X_{14}	...	X_{1n}
	2	X_{21}	X_{22}	X_{23}	X_{24}	...	X_{2n}
	3	X_{31}	X_{32}	X_{33}	X_{34}	...	X_{3n}
	4	X_{41}	X_{42}	X_{43}	X_{44}	...	X_{4n}
	...	...	...	...	...	X_{ij}	...
	m	X_{m1}	X_{m2}	X_{m3}	X_{m4}	...	X_{mn}
Inventory		I_1	I_2	I_3	I_4	I_j	I_n
Demand		D_1	D_2	D_3	D_4	D_j	D_n

3.b Mathematical Modeling

In order to solve an optimization problem, the problem should be modeled as the first step. The objective function should be generated and constraints should be written based on the problem requirements. In the presented problem, some variables are integer numbers while the rest of them are continuous numbers. The objective of this model is to minimize the total cost incurred due to the meeting demand and holding the inventory. As it was discussed in the Problem Introduction, there are also limitations towards meeting demand for each period, suppliers' capacities, supplier selection, and allowed inventory limit.

The following constraints (constraints 2) to 5)) are for meeting the demand (Constraint 2), meeting the suppliers capacity (Constraint 3) and whether there will be a purchase or not (Constraint 5), and inventory limitation for each period (Constraint 4). Through the help of Mixed Integer Programming Modeling, expression 1 and constraints 2 to 5 are derived.

$$1)\ Min \sum_{i=1}^{m} \sum_{j=1}^{n} P_i(X_{ij}).X_{ij} + \sum_{j=1}^{n} h_j.I_j$$

Subject to:

$$2)\ \sum_{i=1}^{m} X_{ij} + I_{j-1} = D_j + I_j \qquad j=1,2,\dots,n$$

$$3)\ y_i.m_i \le X_{ij} \le y_i.M_i \qquad i=1,2,3,\dots,m \qquad j=1,2,\dots,n$$

$$4)\ 0 \le I_j \le r_j$$

$$5)\ y_i \in \{0,1\} \qquad i=1,2,3,\dots,m$$

Expression number 1 is the objective function and the constraints 2 to 5 are the problems constraints. As previously mentioned, suppliers are numerated by variable i which varies from 1 to m and different periods are numerated by variable j which varies from 1 to n. X_{ij}s are the amount of supply from Supplier i in Period j. $P_i(X_{ij})$ is the cost function for supplier i. For simplicity and comparison purposes, linear cost function is used instead which is $P_i(X_{ij}) = P_i$; P_i is the price unit for supplier i. h_j is inventory holding cost per unit per period j. D_j is the demand for period j. m_i and $M_i$are the minimum and maximum allowed amount of supply for supplier i. r_j is the allowed inventory level for period j. y_i

is binary variable which represents presence or absence of a specific supplier in the process of supply.

Expression number 1 is the objective function. In this expression P_i, the supplier price, is multiplied by X_{ij}, which shows the amount of purchase from supplier i in period j; moreover, h_j, inventory holding cost for period j, is multiplied by I_j, which is the inventory at the end of period j. These two factors are added to calculate the total cost.

Constraint 2 shows the amount of purchase from all suppliers in period j plus the inventory delivered from period "j-1" should be equal to the demand for period j plus the inventory at the end of this period.

Constraint 3 shows the capacity restriction for each supplier and whether it will be selected for the problem or not. For example, if y_i is equal to zero it means that supplier y_i is not selected for this problem and there will be no supply from this supplier at all. On the other hand, if y_i is equal to 1, it means that this supplier is selected for this problem; this supplier should supply at least the amount of m_i in each period and this amount of supply should not be more than M_i which is the supplier capacity for this problem.

In constraint number 4, variable I_j, as it was mentioned before, is the Inventory level at the end of each period. This amount in period j should be less than or equal to the specific amount of r_j which is given as the data.

Constraint 5 shows that variable y_i should be binary and only gets 0 or 1value.

In this problem, if all (y_i) were fixed variables, the best feasible solution could be easily found by the help of Linear Programming. In some specific cases that total demand is more than total capacity of all suppliers multiplied by number of periods, there is no feasible solution.

3.c Proposed novel Genetic Algorithm with Binary Representation

In order to solve the proposed model, a GA will be presented. A GA is a heuristic search method constructed on the evolutionary ways of natural selection and genetic science. This method mainly uses Darwin's theory of evolution towards the survival of the fittest chromosomes or specimens and evolutionary biology.

Genetic Algorithms were initially used by John Holland in 1960^{th} in fields other than biology. Afterward, scientists, researchers, and students started using and expanding the GA functions in different fields such as optimization. As a search technique, GAs lead researchers to the exact or an approximate solution. In order to reach this goal, an initial population of chromosomes should be available. Each member of this population or chromosome is made of different genes. Each gene has a specific value. During different steps, these chromosomes evolve and a more desired population is generated. The initial population is generated either randomly or based on coding the current status or available solution of the under study problem. During each iteration, each chromosome is evaluated and its fitness is calculated. By comparing their fitness values and based on a specific pattern defined by the researcher, usually a portion of chromosomes are kept unchanged and rest will be modified. In the modification, different types of genetic operations could be performed on chromosomes. Some researchers use standard operations while other ones may define their own operations based on their demand. The regularly used genetic operations are crossover and mutation. Crossover in GAs is directly inspired from the crossover concept in biology. The parents are cut from similar or different points and the new offspring are generated. Different types of crossovers are used in the GAs such as Single-Point, Two-Point, and Uniform Crossover. The method which is used in this research is Single-Point Crossover.

Mutation is usually a kind of random change which is tested and done separately for each gene of each chromosome in the population. Usually a specific probability is determined and if this chance is met, the gene is mutated. The main goal of mutation is keeping the diversity of the population and preventing the algorithm from being stocked in local optimal solutions; pure crossover could cause this kind of problems.

As previously mentioned, many researchers selected GAs to address similar problems. One or many constraints are selected based on discretion. The selected constraints are manipulated and one or many dummy variables are added to them; these dummy variables help defining the fitness function. The fitness function is defined to evaluate every single chromosome. After evaluation based on defined standards for the problem chromosomes will be accepted or sent to be amended by the help of genetic operators. After the performing the genetic operations, they will be evaluated and again amended. This procedure usually continues up to a specific time limit or number of iterations.

The mathematical formulation which will be used in the GA will be as follow (Expression 6 and constraints 7 to 11).

$$6)\quad Min \sum_{i=1}^{m} \sum_{j=1}^{n} P_i X_{ij} + \sum_{j=1}^{n} h.I_j + R.\sum_{j=1}^{n} \omega_j$$

Subject to:

$$7)\quad \sum_{i=1}^{m} X_{ij} + I_{j-1} + \omega_j = D_j + I_j \qquad j=1,2,\dots,n$$

$$8)\quad y_i.m_i \leq X_{ij} \leq y_i.M_i \qquad i=1,2,3,\dots,m \qquad j=1,2,\dots,n$$

$$9)\quad I_j \leq r_j$$

$$10)\quad y_i \in \{0,1\} \qquad i=1,2,3,\dots,m$$

$$11)\quad 0 \leq \omega_j \qquad j=1,2,\dots,n$$

The fitness function for this genetic algorithm is as follow (Expression 12):

$$12)\qquad F(Y) = R.\sum_{j=1}^{n} \omega_j$$

Expression 6, like Expression 1, is the objective function which will be used in the GA. The same variables are used here, but the term $R.\sum_{j=1}^{n} \omega_j$ is added. R is the penalty parameter which is sufficiently large. ω_j is for balancing the Constraint 7 for period j. The term $R.\sum_{j=1}^{n} \omega_j$ shows the sum of all ω_j for all periods multiplied by R.

Constraint 7 is similar to Constraint 2 and the same variables are used, but the variable ω_j is added to the left side of equation. This dummy variable is added first to help the genetic algorithm to run smooth, and second to help defining the fitness function. Here in period j, if the selected suppliers, by the help of their capacity and the previous period inventory (period "j-1"), are able to meet the demand, the ω_j will be equal to zero; otherwise it takes a non-zero value which first jeopardizes the objective function (Expression 6), and then the fitness function (Expression 12). The reason the dummy variable with a non-negative value is inserted at the right side of equality in Constraint 7 is as follows: if the demand on the right side of the equation is not met, the dummy variable will help it to be met, but it is very costly (in fact impossible) in this model. Constraint 7 could also be written as follows (Constraint 13); in this representation, the relationship between demand, inventory, amount of supply, and dummy variable is shown clearer.

$$13) \qquad \sum_{i=1}^{m} X_{ij} + I_{j-1} + \omega_j - I_j = D_j \qquad j=1,2,\ldots,n$$

Constraint 8 is the same as Constraint 3.

Constraint 9 is the same as Constraint 4.

Constraint 10 is the same as Constraint 5.

Constraint 11 shows the fact that the dummy variable ω_j should be equal to or more than zero.

Expression 12 is the fitness function for the proposed model. The desired value for this function is zero and it is made only in case that the term $\sum_{j=1}^{n} \omega_j$ is equal to zero since R is sufficiently large and positive penalty parameter. R, the penalty coefficient, is a very large number like 10^{10}.

The goal of our algorithm is to minimize the Objective Function (Expression 6) and Fitness Function (Expression 12). If our solution meets all the constraints in the original problem (Constraint 2, 3, 4, and 5), all ω_j will be zero; this condition makes the F(Y) equal to zero. It should be noted that since in the first period there is no inventory, all the demand should be directly met by purchasing from suppliers.

One of the key parts of the GAs is chromosome definition. In addition, genetic operations help the chromosomes evolution. The genetic operators are usually crossover and mutation operations. In the following parts, the way chromosomes are defined and evaluated and the usage of a specific form of Crossover and Mutation in this research will be explained.

Chromosome Definition

Each chromosome in this problem has m genes which is equal to number of Suppliers. The genes are numbered from 1 to m and gene number i corresponds to Supplier i. If a supplier is selected for the supply, the corresponding Gene is 1 and if it is not selected, the corresponding gene is 0. In the other word, gene i is equal to variable "y_i" in the proposed model. In Figure 3 - One sample chromosome is shown.

Gene Number	1	2	3	4	5	6	7	8	9	10
Chromosome	0	1	0	1	1	1	0	0	1	1

Figure 3 - One sample chromosome

In this chromosome, genes 2, 4, 5, 6, 9, and 10 are equal to 1 which means suppliers 2, 4, 5, 6, 9, and 10 will supply, while genes 1, 3, 7, and 8 are equal to 0 which means Suppliers 1, 3, 7, and 8 will not supply.

Chromosome Evaluation

After defining the chromosomes, each chromosome should be evaluated. If they are desirable, they will be kept, otherwise they will be amended. The chromosome Evaluation is done by the help of Expressions 6 to 12, but there is one important point which is the role of each chromosome. Each chromosome determines which y_i should be equal to zero and which y_i should be equal to 1; so the model does not need any more to find the value of each y_i. By the help of this fact, the supplier selection is conducted and just the purchase volumes or amounts from the selected supplier or suppliers should be calculated adding to the Inventory level for each period. In the next step the Fitness Function (Expression 12) should be derived by solving the linear model in Expressions 6 to 11. Expression 6 calculates the final price for each chromosome and Expression 12 calculates the Fitness value for the mentioned chromosome. The only condition for accepting a chromosome is reaching the amount of zero for the corresponding fitness value. If this condition is not met, the amendments by the help of genetic operators (crossover and mutation) should be done. These operators are explained in the next step.

Crossover

The crossover which is used here is a single-point crossover. The first reasons for selecting this scheme is its good performance with the proposed mutation technique; in addition, single point crossover needs less computation time in comparison with the other schemes specifically for very large size problems with large population size. In our initial experimentation it was found that single point crossover performs better in most of the cases. Moreover, this scheme is much simpler to implement than other schemes.. In this technique, two parent chromosomes are selected. The selection procedure is explained in the algorithm. For a chromosome with the length of m, the genes are numbered from 1 (left) to m (right). In Figure 4, one chromosome with m numbers is shown.

1	2	3	4	…	…	…	m-2	m-1	m
				…	…	…			

Figure 4 - One chromosome with m numbered genes

The length of chromosomes for each problem is constant for all the periods and it is equal to number of supplier. The cut point for each chromosome is determined as follow:

- If m is an even number like 6 or 20, the cut point will be exactly after gene number [m/2]. It is shown in Figure 5 - Cut point selection for a chromosome with even number of genes.

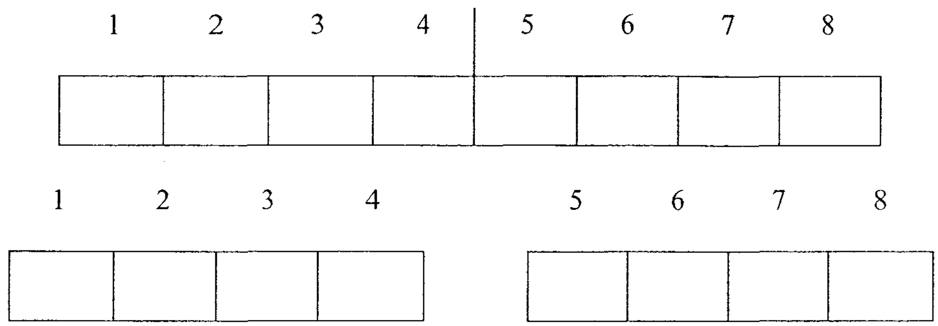


Figure 5 - Cut point selection for a chromosome with even number of genes

- If m is an odd number like 7 or 21, the cut point will be exactly after gene number [m/2]. It is shown in Figure 6 - Cut point selection for a chromosome with odd number of genes.

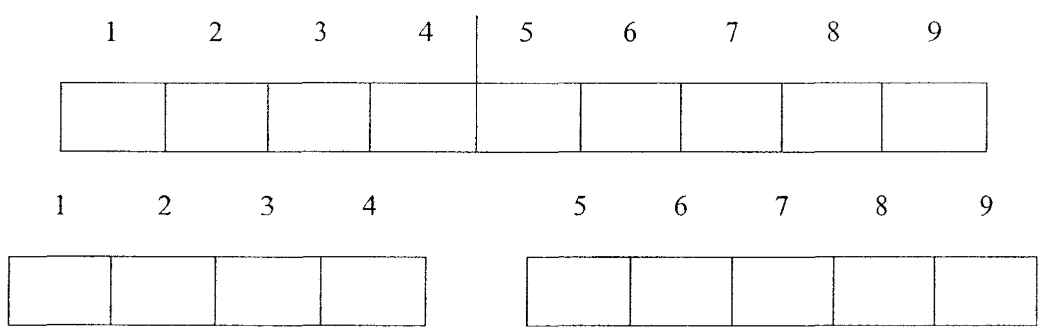
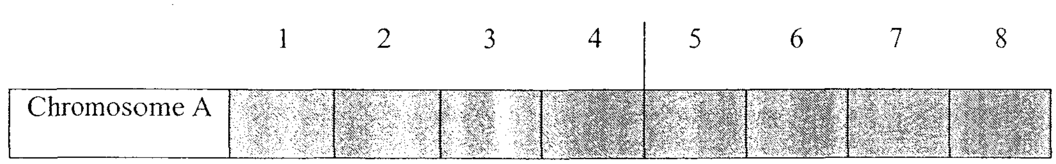


Figure 6 - Cut point selection for a chromosome with odd number of genes

After two chromosomes are cut from the similar cut points, the heads are kept and the tails are interchanged. This is shown in Figure 7.



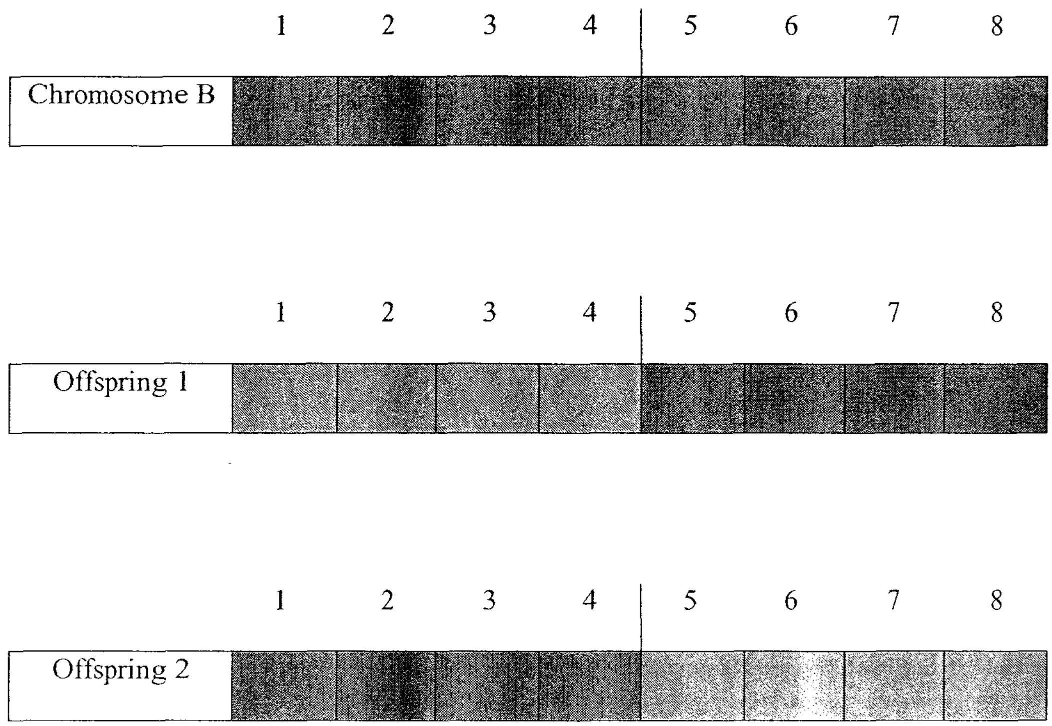


Figure 7 - Single Point Crossover Procedure

The aim of performing the crossover is mainly correcting the existing chromosomes or newly born offspring which are invalid.

Mutation

Mutation is a kind of random change which is imposed on each gene. There are different ways to determine which genes should be mutated. In some cases all genes are targeted for mutation based on a specific probability while in some cases researchers define their own way. In this research a specific way of mutation is used. First two chromosomes are selected and crossover operation is conducted, so two offspring are derived. In the next step, the genes with equal numbers are compared in both chromosomes. If the values are not equal, they will be untouched. If the Values are equal, both are "0" or both are "1", each one of these two genes will be mutated by a given probability which is called P_{mut}. In Figure 8, two chromosomes targeted for mutation are shown.

Gene Number	1	2	3	4	5
Offspring 1	1	1	0	1	0

	1	2	3	4	5
Gene Number					
Offspring 2	0	1	1	0	0

Figure 8 – Two chromosomes targeted for mutation

As we see here, the value of genes 2 and 5 in Offspring 1 are equal to the value of genes 2 and 5 in Offspring 2. Four genes are available here which should face the mutation procedure. Four random real numbers between 0 and 0.99 are generated. For each number less than or equal to P_{mut}, the corresponding gene is mutated. For this example, we assume P_{mut} is equal to 0.18. In Table 2 - mutation sampleis shown.

Table 2 - mutation sample

Offspring No.	Gene No.	Initial Value	Random Number	P_{mut}	Final Value
1	2	1	0.4	0.18	1
2	2	1	0.06		0
1	5	0	0.72		0
2	5	0	0.14		1

Based on the calculation presented in Table 2, both genes 2 and 5 on Offspring 2 are mutated while the genes on Offspring 1 are untouched. In Figure 9 - The Mutation results are shown.

Gene Number	1	2	3	4	5
Offspring 1	1	1	0	1	0

Gene Number	1	2	3	4	5
Offspring 2	0	0	1	0	1

Figure 9 - The Mutation results

This action decreases the probability of having the algorithm stocked in local optimums. Moreover, more research space will be scaled by the algorithm.

In the next step, Programming Algorithm will be explained. By the help of the presented algorithm and programming, the proposed algorithm will be implemented.

3.d Algorithm

The proposed approach was implemented by MATLAB. The pseudo-algorithm is given below:

1) Enter m (number of Suppliers), n (number of Periods), P_i (Supplier Price for each Supplier), h_j (Inventory Holding Cost for each period), D_j (Demand for each period), r_j (allowed inventory limit for each period), popsize (Population Size or number of chromosomes in the model), P_{mut} (Mutation Probability), and φ (Number of iterations). Values m, n, popsize, and φ should be positive integers while the other mentioned ones should be non-negative real numbers. popsize should also be an even number.
2) The Initial Population is randomly generated. Each chromosome has the length of m and the population size for each experiment is defined by the researcher. As it was mentioned in Chromosome Definition, each gene is binary and gets the value of 0 or 1.

3) For all the chromosomes in the Initial Population, we allocate a big amount to the Fitness value, a value like $F_O(Y) = 10^{10}$; O varies from 1 to popsize and represents each chromosome in the current population.

4) R, the Penalty term is set to 10^{10}.

5) For O=1 to popsize, if $F_O(Y) \neq 0$ then solve the optimization problem (Expression 14 to 19) by the help of solver and calculate Expression 20, the fitness Function for each chromosome in the current Population.

14) $Min\, S_O = \sum_{i=1}^{m} \sum_{j=1}^{n} P_i X_{ij} + \sum_{j=1}^{n} h.\, I_j + R.\, \sum_{j=1}^{n} \omega_j$

Subject to:

15) $\sum_{i=1}^{m} X_{ij} + I_{j-1} + \omega_j = D_j + I_j$ $\quad j=1,2,\ldots, n$

16) $y_i.\, m_i \leq X_{ij} \leq y_i.\, M_i$ $\quad i= 1, 2, 3, \ldots, m \quad j=1,2,\ldots, n$

17) $I_j \leq r_j$

18) $y_i \in \{0,1\}$ $\quad i= 1, 2, 3, \ldots, m$

19) $0 \leq \omega_j$ $\quad j=1,2,\ldots, n$

The fitness function for each chromosome is:

20) $F_O(Y) = R.\sum_{j=1}^{n} \omega_j$

6) A part or all of chromosomes for genetic operations are selected as follow:

   a. If less than 10% of chromosomes have $F_O(Y)$ =0, then keep them and perform step 7 on the rest of chromosomes, called pop_two; in this part, if the number of chromosomes less than the mentioned 10% is odd, the chromosome with highest S_O should be assumed as a member of pop_two with regard to step 6) c.

b. If 10% or more of chromosomes have $F_O(Y)$ =0, then keep the first $\left\lfloor \frac{10\%}{2} \right\rfloor . 2$ of chromosomes with lowest S_O and with regard to step 6) c. then perform step 7 on the rest of chromosomes, called pop_two.

c. In case that two or more chromosomes candidate for being a member of pop_two have $F_O(Y)$ =0 and equal S_O, first come first serve policy will be imposed.

d. Sort all the chromosomes based on their Fitness Value from 1 (for the lowest amount) to popsize (for the highest amount). The indexes should also be updated from 1 (for the lowest amount) to popsize (for the highest amount).

7) Genetic operations: the chromosomes which are members of pop_two are selected 2 by 2 in consecutive order. Borisovsky et al (2009) utilized the s-tournament for selecting the parents. Crossover and Mutation are done with previously given explanations. The Mutation Probability (P_{mut}) is given as the data.

8) Make $\varphi = \varphi - 1$. This process monitors the allowed number of Iterations. Different techniques usually exist for termination condition; for example, Borisovsky et al (2009) defined a specific time limit (T) and the overall execution time for their proposed algorithm should be less than T.

9) If $\varphi \geq 0$ then go to step 5 else print all chromosomes with $F_O(Y)$ =0, and print S_O.

10) End

In this section, the problem was introduced and modeled. In addition, a GA for solving the model was proposed. Finally, the programming algorithm was discussed. In the next chapter, Design of Experiments for the proposed algorithm is discussed.

4. Computational results and discussion

Initially in this chapter, the method to generate different problems is explained. Afterwards, different experiments for the introduced GA are conducted. In these experiments, population sizes for different problem sizes are initially determined. Afterwards, proper numbers of iterations for each problem size are determined. Finally, proper mutation probabilities are determined. The problem sizes are defined as follow:

- Small sized problems: Problems up to 5 suppliers and up to 6 periods.
- Medium sized problems: Problems with number of supplier between 6 and 15 and number of periods between 7 and 12.
- Large sized problems: Problems with number of supplier between 16 and 25 and number of periods between 13 and 24.

In this research, it is intended to compare the presented approach with the exact solutions and very large problems were not used. The main idea for this issue is the weakness of MILP solvers for very large problems although GAs could handle these problems. In addition, by using several experimentations and reporting errors, solving several large size problems will be difficult and time consuming. The experimentations in this research will be conducted based on mentioned problem sizes. In this research MILP models are solved by the help of LINGO while GA's are implemented in MATLAB®.

Generating sample problems:

In each problem different constant values exists. Minimum and maximum capacities of each supplier beside the demand for each period are randomly generated by the help of a coding in Matlab. The minimum and maximum capacities should be in a specific range determined by the researcher. The randomly generated demand should be in the following ranges (Constraints 21) and 22)); these ranges do not certify the feasibility of the understudy problem, however disregarding them will certainly lead to an infeasible solution. Besides, the feasibility will finally be verified by the help of MIP approach.

21) $D_j \geq Min\,\{m_i\}$ i= 1, 2, 3, ..., m j=1,2,..., n

22) $D_j \leq \sum_{i=1}^{m} M_i + r_{j-1}$ i= 1, 2, 3, ..., m j=1,2,..., n

Other values such as costs, numbers of periods, and number of suppliers are constant for each problem and selected by the researcher. The prices should not be too variant in order to make a more competitive supplier selection procedure. In case that a few number of suppliers offer prices which are much higher than rest of suppliers with similar capacities, these expensive suppliers will have more chance to be eliminated in this competition. In addition, the inventory holding cost should respect the fact of having a price competitive with different suppliers' prices; this fact makes the options of purchasing from different suppliers and holding inventory for different periods more competitive.

The generated data by the help of Matlab are recorded in an excel file; this file is used as the data input source for both GA and MIP. Table 3 Table 1shows one of randomly generated problems with different number of suppliers and periods for a small sized problem with maximum number of suppliers and periods which is 5 for suppliers and 6 for periods. In appendixes A and B, one sample problem for each of medium and large sized problems is presented.

Table 3 - Sample small sized randomly generated problem

Supplier Number	1	2	3	4	5
Supplier Price	10	13	7	9	16
Min Capacity	738	577	524	152	929
Max Capacity	815	913	547	965	971

Period number	1	2	3	4	5	6
Demands	1970	3264	579	1722	3744	3235
Allowed Inventory Limit (r)	20	25	25	30	25	28
Inventory Holding Cost for each period (h)	20	20	20	20	20	20

Experimentations

In this research, the GA coding was done with the help of MATLAB® 2007a and it was run by CPU AMD TL-60 2GHz and 2 GB RAM. In addition in some experiments, comparisons between GA values and MILP values are done; the MILP values are reached by the help of LINGO 8.0 with the same PC.

The GAs use both exploration and exploitation of the search space. Exploration is done by the help of crossover operators and exploitation is done by the help of mutation. Hansheng and Lishan (1998) defined exploration as generating and checking the diversity by scaling the different areas of the search space while they defined exploitation as amending the undesirable answers in order to get chromosomes with higher fitness.

On basis that the two mentioned operators will help scaling different areas of search space, the performed experiment is very helpful and effective to find the proper strategy. As it was explained before, a mathematical model was initially developed for addressing this problem. By the help of evaluations in different levels, more desirable answers will be selected. During the experiment, the feasibility should always be respected. The data verification is the most significant part in this experiment. Results of different tests for each type and size of problem should be compared. Finally, the best and the most efficient results should be selected.

As it was mentioned before in this research, problems of different sizes are studied. Three different sizes of Small, Medium, and Large problems with different mutation probabilities will be tested. Before this step, different experiments will be developed and conducted to determine the proper amount of population size and number of iterations. Based on the gathered results, the proper mutation probability for each problem size will be proposed. There are different limitations on Student Version of Lingo software for variables and constraints while these limitations are not imposed on the proposed GAs coded by the help of Matlab. The trial version of Lingo only accepts 150 constraints and 300 variables. Moreover, the maximum allowed number of integer variables is 30. The trial version also accepts up to 30 nonlinear expressions (see reference 12).

Liao and Rittscher (2007) or Borisovsky et al (2009) started their approach with a randomly generated initial population. In all the following experiments, the initial populations were also randomly generated and developed in the next steps. The population size remains constant during iterations. The test problems were also randomly generated and solved by MIP techniques as it was explained before; all the test problems are required to be feasible to see the efficiency of the proposed GAs. The number of iterations and the population size for each size of problem are found by developing several experiments. As it was mentioned in each GA, 3 important factors exist:

1) Population size
2) Number of iterations
3) Mutation probability

In this chapter, separate experiments are designed and conducted to find the proper amount for each factor. For the first two factors which are population size and number of iterations, the calculations are done based on the highest possible range for number of suppliers and periods and the results are used for the problems of the same size with less number of suppliers and periods. For the mutation probability, in the first experiments the tests are designed and run with the highest possible ranges for number of suppliers and periods; in the next step, problems with different number of suppliers and periods will be studied.

1) **Population size:** In this research, 3 different sizes of problems are studied: small, medium, and large. In this part, different population sizes will be experimented to find out the best possible size. This fact should also be considered that since the presented chromosome is binary and each gene only takes two possible values of zero or one, the number of possible combinations for a chromosome with the length of m is equal to 2^m. Based on this fact, the possible number of chromosomes for the small sized problems with 5 suppliers is 32, for medium sized problems with 15 suppliers is 32,768 and for large sized problems with 25 suppliers is 33,554,432. Based on this fact, the following population sizes are experimented for each problem size.

   a. Small sized problems: 10, 16, 20, 24 and 30;

   b. Medium sized problems: 20, 30, 40, 50, and 60;

   c. Large sized problems: 60, 70, 80, 90, and 100.

   This fact should be considered that increasing the number of iterations and population size simultaneously will increase the run time of the algorithm. In addition, in case that the number of iterations increases, the chromosomes with non-zero fitness function within different population sizes will have more chance to have a zero fitness function while it is not necessary. Based on the mentioned facts, a balance should be respected between the population size and iterations to find out the effect of each one more precisely. Moreover, although increasing both these factors will probably lead to more precise results, this large numbers will lead to a huge number of calculations which requires a huge amount of resources. Based on discussed issues for small sized problems iterations, the iterations of 10, 20, and 30 will be tried while it is 100, 300, and 500 for medium sized problem. This amount for large sized problems will be 200, 400, and 600. Regarding the mutation probability, Liao and Rittscher (2007) used a random mutation probability between 0.05 and 0.1 and called it PM; in this part of experiment, the same technique for generating the mutation probability is utilized; the only difference is that in this research, this random amount is between 0.02 and 0.1.

In this experiment for each problem size, 10 sample problems will be studied by the mentioned conditions for population size, iteration, and mutation probability. The result of each single run of these experiments will be compared with the same problem which is run with the same mutation probability policy and population size; the only different factor is iterations. For all the problems, a large enough number of iterations (2000) will be selected and run; this experiment will be named as the reference for each test. In case that the result of each test is the same as reference or the error is less than 5%, value "1" will be written in the relevant cell, otherwise it would be "0".

Small Sized Problems:

Table 4 shows the experiment for small sized problems.

Table 4 - Population size experiment for small sized problems

Iteration 10	Population Size				
	10	16	20	24	30
Problem No. 1	0	1	1	1	1
Problem No. 2	0	1	0	1	1
Problem No. 3	1	0	1	1	1
Problem No. 4	0	1	1	1	1
Problem No. 5	1	0	1	0	1
Problem No. 6	0	0	1	1	1
Problem No. 7	0	1	1	0	0
Problem No. 8	1	1	1	0	1
Problem No. 9	0	1	0	1	1
Problem No. 10	1	0	1	0	1

<u>**Iteration 20**</u>	**Population Size**				
	10	16	20	24	30
Problem No. 1	0	0	1	1	1
Problem No. 2	0	1	1	1	1
Problem No. 3	1	0	1	1	1
Problem No. 4	1	1	1	1	1
Problem No. 5	1	1	1	1	1
Problem No. 6	0	0	1	1	1
Problem No. 7	0	1	1	1	1
Problem No. 8	1	0	1	1	1
Problem No. 9	0	1	1	1	1
Problem No. 10	0	1	1	1	1

<u>**Iteration 30**</u>	**Population Size**				
	10	16	20	24	30
Problem No. 1	1	1	1	1	1
Problem No.	1	1	1	1	1

2					
Problem No. 3	0	1	1	1	1
Problem No. 4	1	1	1	1	1
Problem No. 5	1	1	1	1	1
Problem No. 6	0	1	1	1	1
Problem No. 7	1	1	1	1	1
Problem No. 8	1	1	1	1	1
Problem No. 9	1	1	1	1	1
Problem No. 10	1	1	1	1	1

Based on information presented here for iteration 30 and more, the algorithm will lead to almost same answer as the large enough iteration (2000 times in this case) for most of the cases except problems no. 3 and 6 with the population size of 10. For the iteration of 20 times, the population sizes of 10 and 16 did not show an acceptable performance while for 20, 24, and 30, it is acceptable. For iteration of 10, only population size of 30 lead to an acceptable result for all the problems except number 7. Based on presented information here, the population size of 20 is proposed for small size problems.

Medium Sized Problems:

Table 5 shows the experiment for medium sized problems.

Table 5 - Population size experiment for medium sized problems

Iteration 100	Population Size				
	20	30	40	50	60
Problem No. 1	0	1	0	1	0
Problem No. 2	0	0	1	0	1
Problem No. 3	1	0	0	1	0
Problem No. 4	0	0	1	0	1
Problem No. 5	0	0	0	0	1
Problem No. 6	0	1	0	1	0
Problem No. 7	0	0	0	0	1
Problem No. 8	0	1	1	0	1
Problem No. 9	0	0	1	0	1
Problem No. 10	1	0	1	1	0

<u>**Iteration 300**</u>	**Population Size**				
	20	30	40	50	60
Problem No. 1	0	0	1	1	1
Problem No. 2	0	1	1	1	1
Problem No. 3	1	0	1	1	1
Problem No. 4	1	1	1	1	1
Problem No. 5	0	0	1	1	1
Problem No. 6	1	0	1	1	1
Problem No. 7	1	0	1	1	1
Problem No. 8	0	1	1	1	1
Problem No. 9	1	0	1	1	1
Problem No. 10	0	0	1	1	1

<u>**Iteration 500**</u>	**Population Size**				
	20	30	40	50	60
Problem No. 1	0	1	1	1	1
Problem No.	1	0	1	1	1

2					
Problem No. 3	1	1	1	1	1
Problem No. 4	0	1	1	1	1
Problem No. 5	1	0	1	1	1
Problem No. 6	0	1	1	1	1
Problem No. 7	1	1	1	1	1
Problem No. 8	1	1	1	1	1
Problem No. 9	0	1	1	1	1
Problem No. 10	1	1	1	1	1

Based on information presented here for iteration 500 and more, the algorithm will lead to almost the same answer as the large enough iteration (2000 times in this case) for most of the cases. For the iteration of 300 times, the population sizes of 20 and 30 did not show acceptable performances while for the rest, it is acceptable. For iteration of 100, none of the population sizes led to an acceptable result. Based on presented information here, the population size of 40 is proposed for medium size problems.

Large Sized Problems:

Table 6 shows the experiment for large sized problems.

Table 6 - Population size experiment for large sized problems

Iteration 200	Population Size				
	60	70	80	90	100
Problem No. 1	1	0	0	0	1
Problem No. 2	0	1	1	0	0
Problem No. 3	1	1	1	1	0
Problem No. 4	0	0	0	1	1
Problem No. 5	0	0	0	1	0
Problem No. 6	0	0	0	1	0
Problem No. 7	1	0	0	0	1
Problem No. 8	0	1	1	0	1
Problem No. 9	0	1	0	1	1
Problem No. 10	1	1	0	1	0

Iteration 400	Population Size				
	60	70	80	90	100
Problem No. 1	1	1	1	1	1
Problem No. 2	0	0	1	1	1
Problem No. 3	1	1	1	1	1
Problem No. 4	0	1	1	1	1
Problem No. 5	0	0	1	1	1
Problem No. 6	0	0	1	1	1
Problem No. 7	0	1	1	1	1
Problem No. 8	1	1	1	1	1
Problem No. 9	0	1	1	1	1
Problem No. 10	0	0	1	1	1

Iteration 600	Population Size				
	60	70	80	90	100
Problem No. 1	1	1	1	1	1
Problem No.	1	1	1	1	1

2					
Problem No. 3	1	1	1	1	1
Problem No. 4	1	1	1	1	1
Problem No. 5	1	1	1	1	1
Problem No. 6	1	1	1	1	1
Problem No. 7	1	1	1	1	1
Problem No. 8	1	1	1	1	1
Problem No. 9	1	1	1	1	1
Problem No. 10	1	1	1	1	1

Based on information presented here for iteration 600 and more, the algorithm will lead to almost the same answer as the large enough iteration (2000 times in this case) for all of the cases. For the iteration of 400 times, the population sizes of 60 and 70 did not show an acceptable performance while for the rest, it is acceptable. For iteration of 200, none of the population sizes led to an acceptable result. Based on presented information here, the population size of 80 is proposed for large size problems.

2) **Number of iterations:** As it was mentioned before, 3 different sizes of problems are studied here. In order to find out the best number of iteration for each problem size, different experiments are conducted as follow. 10 different problems of each size are run with large enough number of iterations (it is 1000 in this case). The number of chromosomes with non-zero fitness function will decrease by passing iterations and it will approximately reach a constant amount. The best number of iterations is selected based on these values on the spot that amount of chromosomes with non-zero fitness function reaches an acceptable amount which is equal to or less than 60% of total population size. This amount is determined based on researcher's discretion through conducting several experiments and by the study of different manners of GA's for the similar problems. Regarding the mutation probability, again Liao and Rittscher (2007) technique for generating the mutation probability is utilized. Liao and Rittscher (2007) also used the number of iterations in their research as the termination condition for the GA. The population size used here is based on the information from the previous experiment. The same as previous experiments, these experiments are conducted with the highest possible amount of population size and iteration number for each problem size; the results of these experiments are extended to the problems with the same size.

Small Sized Problems:

Table 7 shows the relaxation iteration level and corresponding number of chromosomes with non-zero fitness function beside the mentioned 60% of chromosomes with non-zero fitness function which is 12 for the small sized problems. Based on extracted information and presented graph (Figure 10), at the worst condition algorithm reaches the acceptable condition for problem 4 at the end of iteration number 13. In order to have more confidence, a higher number should be selected and here 20 is proposed by the researcher.

Table 7 - Relaxation iteration level and corresponding number of chromosomes with non-zero fitness function for small sized problems

Problem Number	Final iteration relaxation number	Number of chromosomes with non-zero fitness function at relaxation level	Number of iteration reaching 12 as number of solutions with non-zero fitness function
1	47	1	5
2	9	1	1
3	27	1	7
4	14	11	13
5	39	1	1
6	5	1	1
7	79	3	11

Problem Number	Final iteration relaxation number	Number of chromosomes with non-zero fitness function at relaxation level	Number of iteration reaching 12 as number of solutions with non-zero fitness function
8	66	1	8
9	61	1	2
10	11	1	12

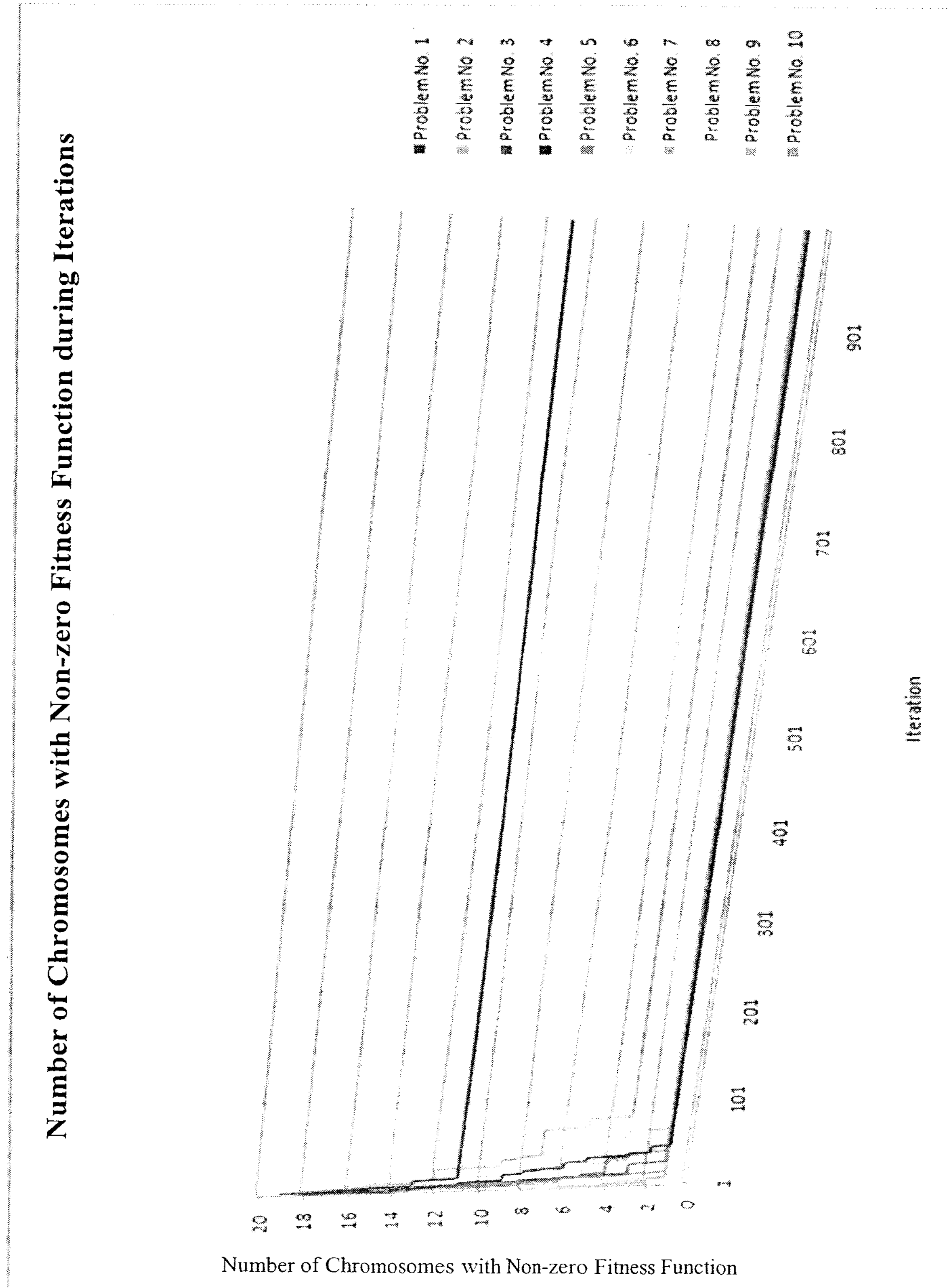


Figure 10 - Number of chromosomes with non-zero fitness function during iterations for small sized problems

Medium Sized Problems:

Table 8 shows the relaxation iteration level and corresponding number of chromosomes with non-zero fitness function beside the mentioned 60% of chromosomes with non-zero fitness function which is 24 for the medium sized problems. Based on extracted information and presented graph (Figure 11), at the worst condition algorithm reach the acceptable condition for problem 1 at the end of iteration number 166. In order to have more confidence, a higher number should be selected which could be 200 here; in order to have more precise answers, 300 is selected by the researcher for this size of problems.

Table 8 - Relaxation iteration level and corresponding number of chromosomes with non-zero fitness function for medium sized problems

Problem Number	Final iteration relaxation number	Number of chromosomes with non-zero fitness function at relaxation level	Number of iteration reaching 24 as number of solutions with non-zero fitness function
1	471	8	166
2	30	1	1
3	73	1	1
4	107	1	5
5	795	2	93
6	74	1	6
7	372	1	9

Problem Number	Final iteration relaxation number	Number of chromosomes with non-zero fitness function at relaxation level	Number of iteration reaching 24 as number of solutions with non-zero fitness function
8	64	1	1
9	951	9	152
10	65	1	2

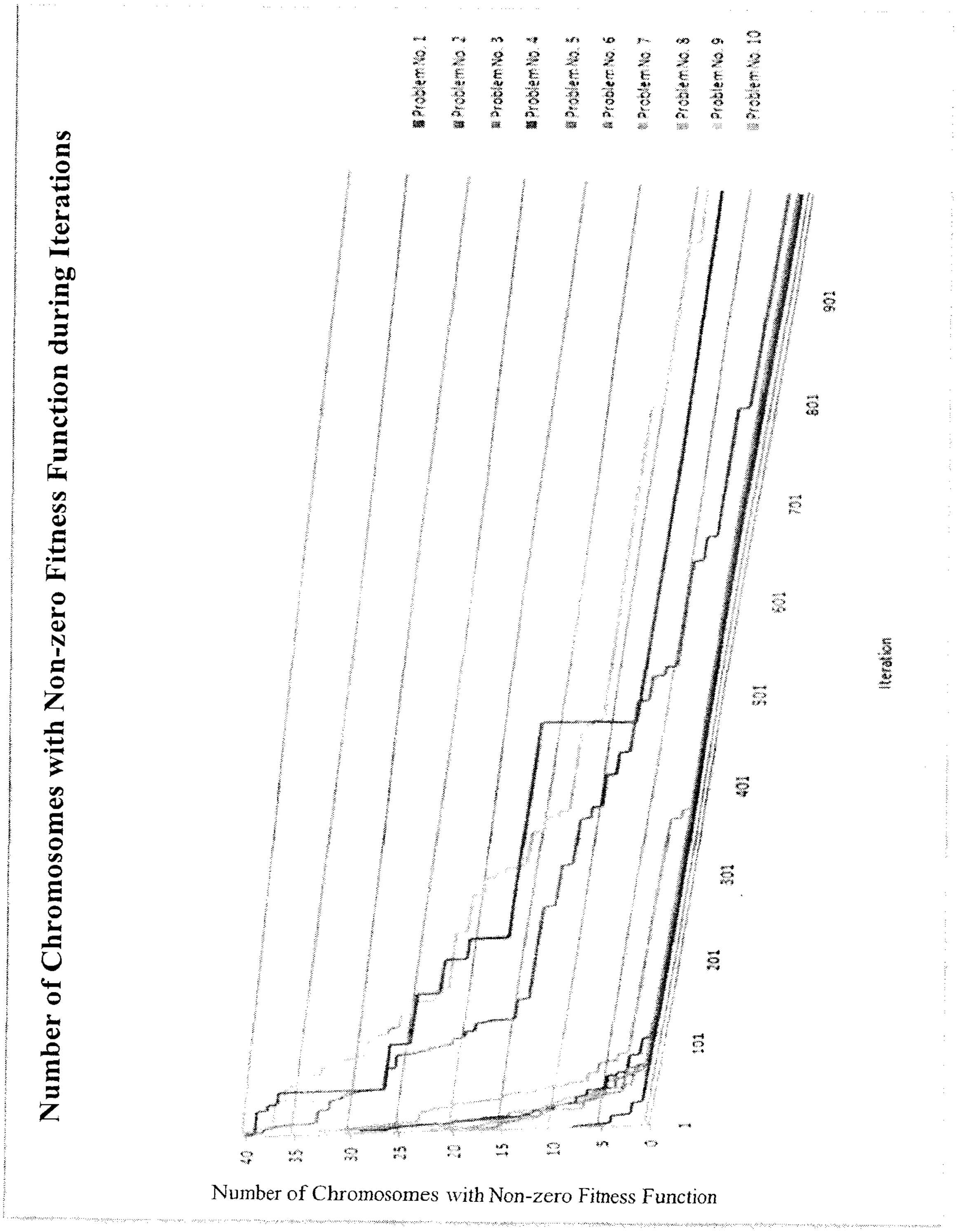


Figure 11 - Number of chromosomes with non-zero fitness function during iterations for medium sized problems

Large Sized Problems:

Table 9 shows the relaxation iteration level and corresponding number of chromosomes with non-zero fitness function beside the mentioned 60% of chromosomes with non-zero fitness function which is 48 for the large sized problems. Based on extracted information and presented graph (Figure 12), at the worst condition algorithm reach the acceptable condition for problem 7 at the end of iteration number 333. In order to have more confidence, a higher number should be selected which is 400 here.

Table 9 - Relaxation iteration level and corresponding number of chromosomes with non-zero fitness function for large sized problems

Problem Number	Final iteration relaxation number	Number of chromosomes with non-zero fitness function at relaxation level	Number of iteration reaching 48 as number of solutions with non-zero fitness function
1	34	1	1
2	39	1	1
3	35	1	1
4	958	7	136
5	654	5	297
6	46	1	1
7	462	11	333
8	35	1	1

Problem Number	Final iteration relaxation number	Number of chromosomes with non-zero fitness function at relaxation level	Number of iteration reaching 48 as number of solutions with non-zero fitness function
9	31	1	1
10	281	1	11

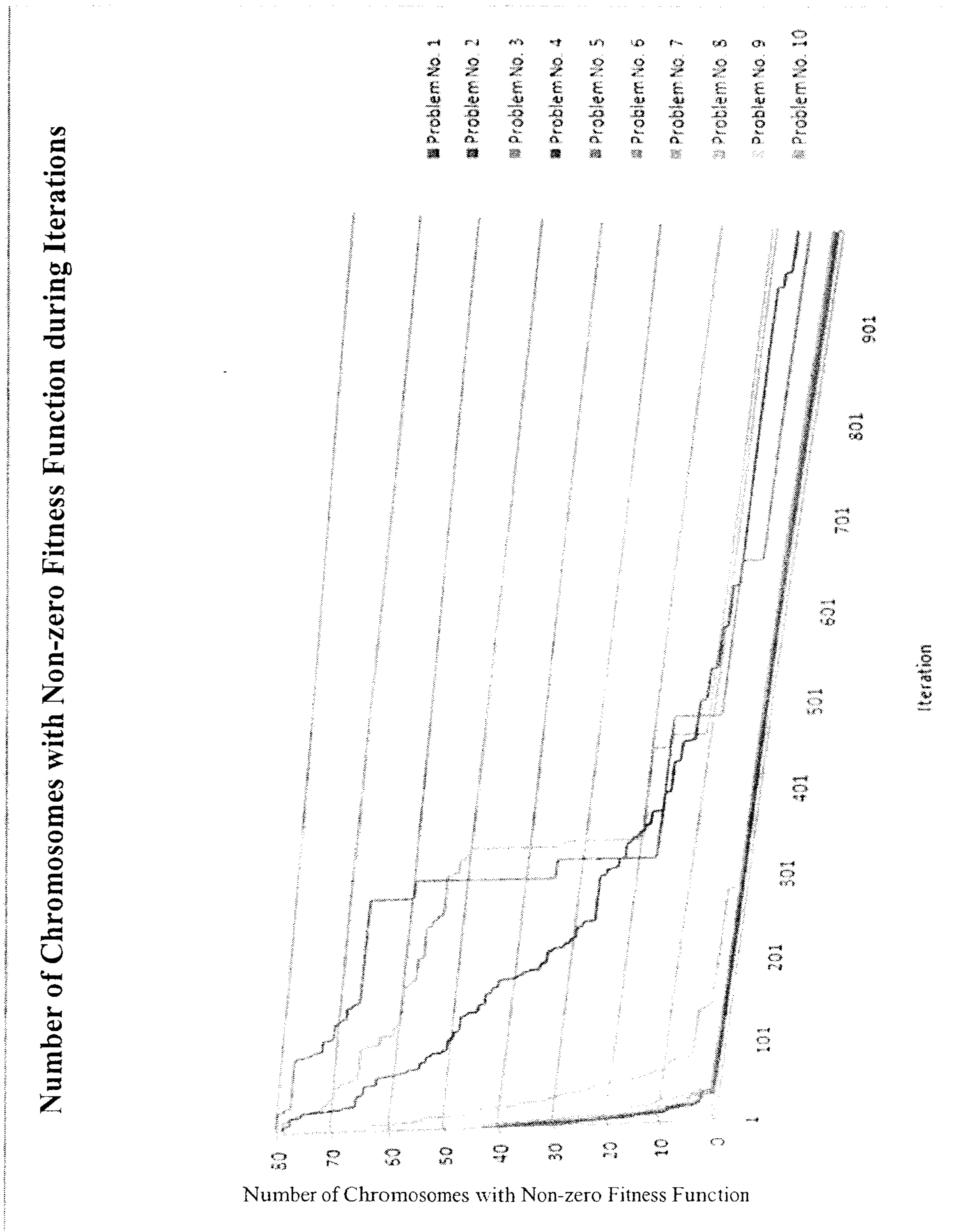


Figure 12 - Number of chromosomes with non-zero fitness function during iterations for large sized problems

3) **Mutation probability:** In this part of the research, problems of different sizes are again studied. The same as before, these three different sizes are Small, Medium, and Large problems with different mutation probabilities. The goal is to find the best possible probability for each problem size. For the employed crossover scheme, researchers usually use mutation probabilities between 0.02 and 0.08. In this section, proper mutation probabilities for further experiments are initially determined by the help of different sets of experiments; the mutation probabilities for the first set of experiments are 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, and 0.10. In these experiments, problems of different sizes are generated by the highest number of suppliers and periods and the result are used for the problems of the same size with different number of suppliers and periods. The decision factor for selecting proper mutation probabilities for different sizes of problems is mean of errors. The error is calculated by the help of (Formula 23);

$$23) \qquad Error(\%) = \frac{(GA\ Value - MIP\ Value)}{MIP\ Value} \times 100$$

The results for determining the candidate mutation probabilities for further experiments are shown in Table 10.

Table 10 - Error (%) for Different Mutation Probabilities for Different Problem Sizes

Small Sized Problem											
Problem No.	Error (%) with respect to mutation probability of										
	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10	0.11
1.00	14.63	0.87	0.55	4.80	0.87	11.35	14.63	0.87	4.68	0.87	5.60
2.00	3.81	3.81	10.71	12.42	3.81	10.17	15.71	3.81	4.25	3.30	15.71
3.00	17.24	3.22	10.33	10.36	3.22	3.45	10.54	3.22	12.02	3.22	4.35
4.00	4.20	6.39	8.96	8.03	6.39	4.56	6.98	6.39	5.20	6.39	7.68
5.00	8.89	5.90	10.14	14.65	3.65	7.59	7.32	3.65	5.36	5.90	6.12
Mean of Errors (%)	9.75	**4.04**	8.14	10.05	**3.59**	7.42	11.04	**3.59**	6.30	**3.94**	7.89

Medium Sized Problem											
Problem No.	Error (%) with respect to mutation probability of										
	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10	0.11
1.00	7.23	3.45	3.45	7.62	1.48	4.35	5.30	6.79	5.63	2.84	7.12
2.00	4.32	1.29	4.71	3.50	1.27	3.95	4.61	0.95	4.38	1.02	3.89
3.00	8.11	3.43	5.63	3.47	3.65	7.24	7.14	3.71	9.11	3.50	5.31
4.00	4.41	1.66	4.95	6.30	1.70	4.62	2.47	2.78	10.19	2.02	1.98
5.00	3.22	3.81	7.01	5.63	3.81	7.13	8.69	6.81	14.21	8.73	12.69
Mean of Errors (%)	5.46	**2.73**	5.15	5.30	**2.38**	5.46	5.64	**4.21**	8.70	**3.62**	6.20

Large Sized Problem											
Problem No.	Error (%) with respect to mutation probability of										
	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10	0.11
1.00	6.19	3.22	5.34	6.32	2.52	4.32	6.22	2.45	4.65	3.33	4.69
2.00	6.34	2.35	7.89	4.50	4.60	4.60	4.75	4.80	13.36	5.35	11.02
3.00	3.01	3.14	7.34	7.21	4.85	11.33	5.34	3.27	6.14	4.68	6.35
4.00	5.25	5.31	5.05	5.21	3.56	5.98	6.32	7.84	6.89	5.76	5.42
5.00	9.37	7.45	10.20	8.69	6.13	8.95	9.37	5.87	8.21	4.35	7.24
Mean of Errors (%)	6.03	**4.29**	7.16	6.39	**4.33**	7.04	6.40	**4.85**	7.85	**4.69**	6.94

Based on conducted experiments, proper mutation probabilities for this research are 0.02, 0.05, 0.08, and 0.1 since these mutation probabilities generated lowest mean errors among all the utilized mutation probabilities. Based on the gathered results in the next experiments, the proper mutation probability for each problem size will be proposed. For each problem size, 20 different problems are studied; 10 of these problems are with highest allowed number of suppliers and periods for each problem size and the rest are with different number of suppliers and periods allowed for each problem size. Results of previous experiments for number of iterations and population size are utilized in this section. During the experiments, the objective function values derived by the GA and MIP are compared with each other and the error is calculated; finally the mean and standard deviation for the errors are calculated. The formula for calculating the error is again formula 23).

The proper mutation probability is selected for each problem based on the presented data. Comparison between the result of GAs solution and the result of MIP solution for small sized problems shows that the result of GAs is also acceptable for bigger problems. As it was mentioned before, there are different limitations on Student Version of Lingo software for variables and constraints while these limitations are not imposed on the proposed GAs coded by the help of Matlab.

Liao and Rittscher (2007) or Borisovsky et al (2009) started their approach with a randomly generated initial population. In all the following experiments, the initial populations were also randomly generated and developed in the next steps. The same as before, population size remains constant during iterations. The test problems were also randomly generated and solved by MIP techniques; all the test problems are required to be feasible to see the efficiency of the proposed GAs. The number of iterations and the population size for each size of problem were found by developing several experiments; these experiments were explained in previous parts of the research.

Small Sized Problem

As it was mentioned before, small sized problems are problems with number of supplier of 4 or 5 and number of periods between 4 and 6. In this section, 10 different problems with 5 suppliers and 6 periods and 10 different problems with different number of suppliers and periods in the mentioned range are studied. One of these problems limitations are shown in Table 11.

Table 11 - Suppliers Specifications

Supplier Number	Capacity Lower Bound	Capacity Upper Bound	Price
1	53	145	10
2	48	95	13
3	110	250	7
4	46	100	9
5	30	95	16

The demands for each period, the inventory holding cost for each period, and allowed inventory level for this problem are shown in Table 12.

Table 12 - Periods Information

Period number	1	2	3	4	5	6
Demands	350	420	190	130	200	310
Allowed Inventory Limit (r)	20	25	25	30	25	28
Inventory Holding Cost for each period (h)	20	20	20	20	20	20

During this section, the problem is solved with different mutation probabilities which are 0.02, 0.05, 0.08, and 0.1. The population size for this size of problem is 20 and the iteration for small sized problems is 20. For each problem, the program was run and the gathered data are used to calculate the errors. The results of test for small sized problem are shown in Table 13. In this table, three important factors as the result of the implemented GA's are shown. These factors are the error percentages, the standard deviations of errors, and means of errors. Kumar et al. (2000) used the difference between the GA's solution and LP solution to validate their results towards the proper answer. The results gathered by GAs may not be optimal and a comparison is done between GAs solution and MIP solution. The average run time of the proposed GA for small sized problems is approximately 1 minute.

Table 13 - Results for small sized problems

Problem No.	Number of Suppliers	Number of Periods	Error (%) with respect to mutation probability of			
			0.02	0.05	0.08	0.1
1	5	6	6.39	6.39	6.39	6.39
2	5	6	8.45	8.45	7.29	18.03
3	5	6	5.90	3.65	3.65	5.90
4	5	6	3.30	3.30	3.42	9.80
5	5	6	2.32	8.38	2.32	8.38
6	5	6	0.87	0.87	0.87	1.45
7	5	6	15.71	8.38	8.38	15.71
8	5	6	17.24	17.24	17.24	17.24
9	5	6	3.81	3.81	3.81	5.30
10	5	6	14.63	14.63	14.63	15.95
11	5	6	16.80	7.27	7.27	7.27
12	5	6	15.02	5.55	5.55	15.02
13	5	5	6.14	6.14	6.14	6.14
14	5	5	17.23	7.32	7.32	7.32
15	5	4	15.73	8.40	5.20	10.40
16	4	6	5.72	5.72	5.72	5.72
17	4	6	8.55	8.55	8.55	8.55
18	4	5	14.30	8.42	8.42	16.30
19	4	5	13.81	4.22	4.00	13.81
20	4	4	10.21	8.32	5.51	5.51
Standard Deviation of Errors			5.43	3.61	3.74	4.81
Mean Percentage Error			10.11	7.25	6.58	10.01

Figure 13 shows the mean of errors for different mutation probabilities for small sized problems in 20 different studied problems.

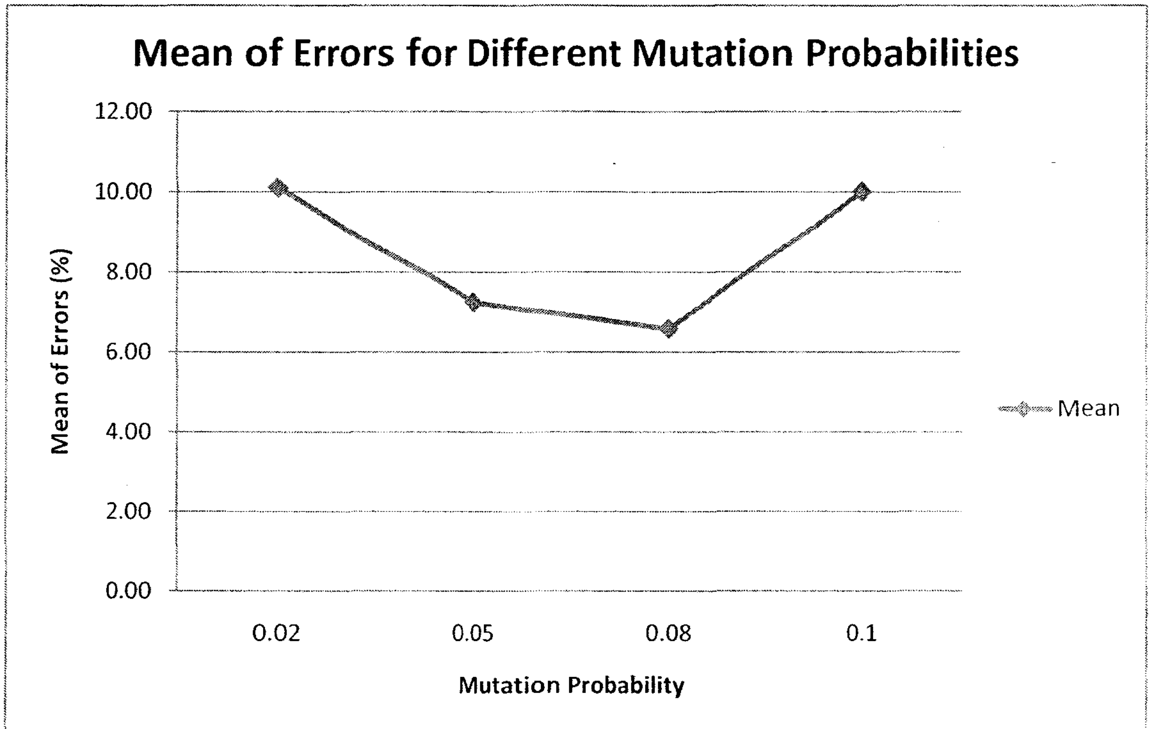


Figure 13 - Mean of errors for different mutation probabilities for small sized problems

Based on presented graph and information, in some cases the results are the same or very close for each mutation probability (for example problem no. 1). In some cases, the results significantly change based on changing the mutation probability (for example problems number 2 and 3). The decision factor for selecting the best mutation probability for the presented experiments is the mean of calculated error percentages. In this part, the mean of errors for probability of 0.08 is 6.58 which is less than other cases. Based on the presented data here, the best mutation probability for the small sized problems in this research is 0.08; in problems 2, 15, and 19, this mutation probability is the uncompetitive one which leads to the best answer for the GA's. In some cases (for example problems 1, 3, 5, 17, and 18), this mutation probability leads to the best answer simultaneously with other probabilities. For problem 4, this amount wins the second place as the best mutation probability. Since the other probabilities are not as frequent as this one and based on the calculated mean, the researcher proposes this amount as the proper mutation probability for small sized problems. Any change in this amount will cause increasing the total cost and consequently the error percentage. In addition, the standard deviations for these problems are slightly alternating in different cases and could not be considered as an important decision making factor for this problem size. It is important to consider the fact that the standard deviation of errors for mutation probability of 0.08 wins the second lowest place of these values.

Medium Sized Problem

As it was mentioned before, medium sized problems are problems with number of supplier between 6 and 15 and number of periods between 7 and 12. In this section, 10 different problems with 15 suppliers and 12 periods and 10 different problems with different number of suppliers and periods in the mentioned range are studied. During this section like the previous section, the problem is solved with different mutation probabilities which are 0.02, 0.05, 0.08, and 0.1. The population size for this size of problem is 40. The iteration for medium sized problems is 300. For each problem, the program was run and the gathered data are used to calculate the errors. The results of test for medium sized problem are shown in Table 14. In this table, three important factors as the result of the implemented GA's are shown. These factors are the error percentages, the standard deviations of errors, and means of errors. The average run time for medium sized problems is approximately 5 minutes.

Table 14 - Results for medium sized problems

Problem No.	Number of Suppliers	Number of Periods	Error (%) with respect to mutation probability of			
			0.02	0.05	0.08	0.1
1	15	12	3.45	1.48	6.79	2.84
2	15	12	6.64	9.60	9.75	8.02
3	15	12	3.23	4.01	4.66	4.07
4	15	12	1.29	1.27	0.95	1.02
5	15	12	2.34	1.91	2.76	2.34
6	15	12	3.96	6.28	9.51	3.41
7	15	12	3.43	3.65	3.71	3.50
8	15	12	2.69	1.71	2.30	1.99
9	15	12	1.26	1.90	4.00	8.12
10	15	12	1.66	1.70	2.78	2.02
11	15	10	5.23	3.78	3.19	3.78
12	14	12	6.33	5.96	7.59	5.45
13	11	10	3.81	3.81	6.81	8.73
14	10	8	4.95	13.07	9.41	2.88
15	9	11	5.23	9.58	16.15	7.31
16	9	7	7.46	7.46	7.46	7.46
17	8	10	5.30	8.54	7.35	7.35
18	7	12	2.09	2.09	2.09	2.09
19	7	9	4.42	8.69	8.69	8.69
20	6	10	5.23	3.78	3.19	3.78
Standard Deviation of Errors			1.75	3.35	3.59	2.54
Mean Percentage Error			4.00	5.01	5.96	4.74

Figure 14 shows the mean of errors for different mutation probabilities for medium sized problems in 20 different studied problems.

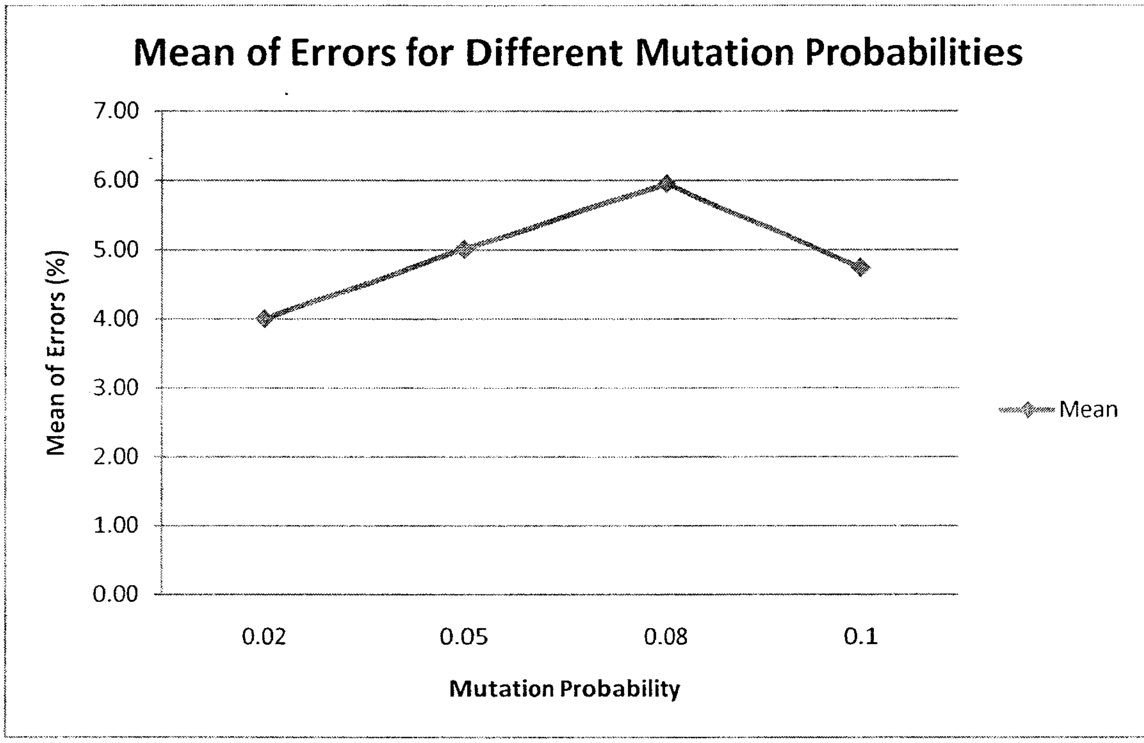


Figure 14 - Mean of errors for different mutation probabilities for medium sized problems

Based on presented graph and information, in some cases the results are the same or very close for each mutation probability. In some cases, the results significantly change based on changing the mutation probability (for example problems number 1, 2, 6, 9, 14, and 15). The decision factor for selecting the best mutation probability for the presented experiments is the mean of calculated error percentages. In this part, the mean of errors for probability of 0.02 is 4.00 which is less than other cases. Based on the presented data here, the best mutation probability for the medium sized problems in this research is 0.02; in problems 2, 3, 7, 9, 10, 14, 15, 17, and 19, this mutation probability leads to the best answer for the GA's and for problem 5, this amount wins the second place as the best mutation probability. Since this probability leads to the lowest mean of errors and the other probabilities are not as frequent as this one, the researcher proposes this amount as the proper mutation probability for medium sized problems. Any change in this amount will cause increasing the total cost and consequently the error percentage. In addition, the standard deviations of errors in these problems are not as close as those of small sized problems. Based on information presented here, a tendency for generating less deviated errors for mutation probability of 0.02 is also more vivid in comparison with other mutation probabilities.

Large Sized Problem

As it was mentioned before, large sized problems are problems with number of supplier between 16 and 25 and number of periods between 13 and 24. In this section, 10 different problems with 25 suppliers and 24 periods and 10 different problems with different number of suppliers and periods in the mentioned range are studied. During this section like the previous sections, the problem is solved with different mutation probabilities which are 0.02, 0.05, 0.08, and 0.1. The population size for this size of problem is 80. The iteration for large sized problems is 400. For each problem, the program was run and the gathered data are used to calculate the errors. The results of test for large sized problem are shown in Table 16. In this table, three important factors as the result of the implemented GA's are shown. These factors are the error percentages, the standard deviations of errors, and means of errors. Large size problems take one average more time because of huge number of chromosomes as well as calculation. The instances take between 25 to 30 minute. The results of GA and MIP for one of the large sized problems are shown in Table 15.

Table 15 - GA and MIP results for a sample large sized problem

Problem No.	Number of Suppliers	Number of Periods	Mutation Probability	GA Value	MIP Value	Error (%) with respect to mutation probability of
6	25	24	0.02	1,078,706.00	1,045,050.00	3.22
			0.05	1,071,360.00		2.52
			0.08	1,070,694.00		2.45
			0.10	1,079,882.00		3.33

Table 16 - Results for large sized problems

Problem No.	Number of Suppliers	Number of Periods	Error (%) with respect to mutation probability of			
			0.02	0.05	0.08	0.1
1	25	24	6.11	7.00	14.33	12.27
2	25	24	7.68	5.33	8.03	11.01
3	25	24	5.31	3.56	7.84	5.76
4	25	24	7.45	6.13	5.87	4.35
5	25	24	8.31	5.43	5.93	5.93
6	25	24	3.22	2.52	2.45	3.33
7	25	24	6.18	3.18	4.64	6.44
8	25	24	5.70	5.12	9.37	3.95
9	25	24	5.70	10.29	10.97	9.51
10	25	24	7.82	6.35	7.59	5.71
11	25	20	4.96	7.88	6.83	5.82
12	23	15	8.04	4.18	7.93	7.06
13	22	17	6.73	8.25	5.54	6.02
14	20	24	7.78	5.35	14.76	9.78
15	20	15	7.84	6.42	13.59	9.85
16	19	22	5.03	2.73	4.84	4.45
17	19	18	2.35	4.60	4.80	5.35
18	18	24	3.14	4.85	3.27	4.68
19	17	15	7.11	7.70	3.99	3.25
20	16	24	6.88	0.95	0.95	1.57
Standard Deviation of Errors			1.71	2.17	3.75	2.75
Mean Percentage Error			6.17	5.39	7.18	6.30

Figure 15 shows the mean of errors for different mutation probabilities for large sized problems in 20 different studied problems.

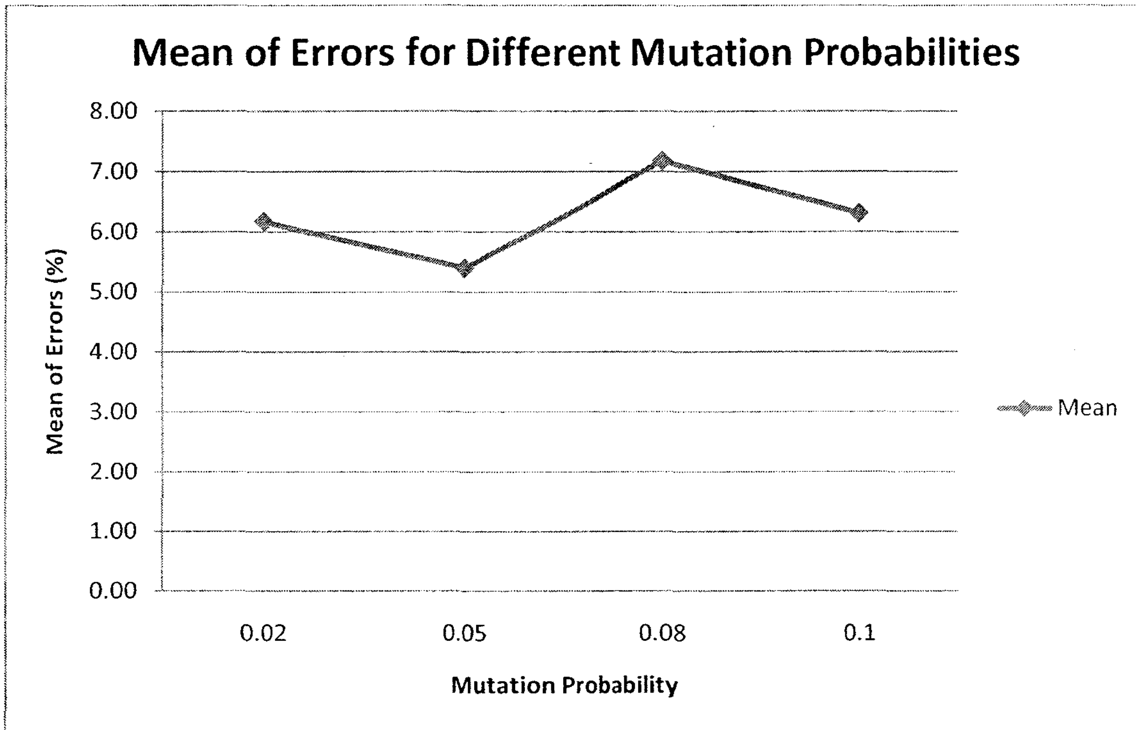


Figure 15 - Mean of errors for different mutation probabilities for large sized problems

Based on presented graph and information, in some cases the results are the same or very close for each mutation probability (for example problem number 5). In some cases, the results significantly change based on changing the mutation probability (for example problems number 1, 2, 3, 4, 14 and 20). The decision factor for selecting the best mutation probability for the presented experiments is the mean of calculated error percentages. In this part, the mean of errors for probability of 0.05 is 5.39 which is less than other cases. Based on the presented data here, the best mutation probability for the large sized problems in this research is 0.05; in problems 2, 3, 5, 7, 8, 12, 14, 15, and 16, this mutation probability leads to the best answer for the GA's. Since the other probabilities do not show a consistent acceptable performance as frequent as this one, the researcher proposes this amount as the proper mutation probability for large sized problems. Any change in this amount will cause increasing the total cost and consequently the error percentage.

5. Conclusions

Genetic algorithm based approaches are very popular and successful in the areas of combinatorial optimization. Researchers have used and proposed GAs in different fields of supply chain management including supplier selection problems. In this research, a novel GA was encoded to address the supplier selection problem with multiple capacitated suppliers in different periods. The objective of the proposed algorithm was to minimize the total cost incurred due to purchasing the products and holding inventory in different periods for three different problem sizes which are small, medium, and large. In this research, linear function was used in order to compare the presented approach by the researcher with respect to the exact approach. In case of existence of concave function, the approach presented in Chauhan et al. instead of using LP could be used in order to evaluate fitness function. In each GA, three role playing factors exist; these three factors are population size, number of iterations, and mutation probability. As the first experimental step in this research, the proper population size was determined by the help of several experiments and comparing them to a reference experiment for each problem size. Afterwards, several experiments were conducted to discover the proper number of iteration for each problem size. Finally, different mutation probabilities for supplier selection problems of various complexities were tested and the effect of various mutation probabilities was studied on these problems. Based on the reported errors on each problem and mean of errors for each mutation probability for all sample problems, the proper mutation probabilities were selected. Proposed mutation probabilities lead the researcher to find the answer of the model with less error. The experimentation results on the randomly generated problems show that GAs are very effective towards solving different sizes of under study supplier selection problems including large problems and this technique could be extended to the problems of similar sizes or even larger sizes with acceptable ranges of results. The presented approach is capable of handling similar supplier selection problems with more constraints and larger data, especially real world supplier selection problems; however it may not guarantee to reach a global optimum solution for very large size problems in a very short period of time. The GA solution to very large problems can be used as a starting solution for the Branch-and-Bound algorithm if the optimal solution is of prime importance.

6. Future Studies:

The research presented in this thesis can be improved in many ways. The problem which was addressed here had a linear objective function and linear constraints. In many cases in real world problems, researchers face problems with non-linear objective functions and/or constraints. Besides these items, a set of suppliers may face different customers with different demands. Each supplier may supply different products and each customer may require single or multiple products. In many cases, there is a chance that customers may have to handle the transportation themselves; based on this restriction, they should also minimize the transportation cost. Ordering products from different suppliers may also impose fixed ordering costs for each supplier. It is also possible to impose restrictions on selecting specific suppliers together or not selecting specific suppliers together; for instance, due to transportation limitations and location of a subset of specific suppliers, it is not possible to purchase more than or less than a specific amount from that specific subset of suppliers. Besides, service level could impose more restrictions on the model.

References

1. Arostegui Jr. M.A., Kadipasaoglu S.N. and Khumawala B.M, 2006. "An empirical comparison of Tabu Search, Simulated Annealing, and Genetic Algorithms for facilities location problems", International Journal of Production Economics, **103 (2),** 742-754.
2. Benton W.C., 1991. "Quantity discount decisions under conditions of multiple items, multiple suppliers, and resource limitations", International Journal of Production Research, **29,** 1953–1961.
3. Biegler L.T. and Grossmann I.E., 2004. "Retrospective on optimization', Computers & Chemical Engineering, **28,** 1169-1192.
4. Borisovsky P., Dolgui A. and Eremeev A., 2009. "Genetic algorithms for a supply management problem: MIP-recombination vs greedy decoder", European Journal of Operational Research, **195**, 770-779.
5. Burke G.J., Carrillo J. and Vakharia A.J., 2007. "Single versus multiple supplier sourcing strategies", European Journal of Operational Research, **182**, 95-112.
6. Burke G.J., Carrillo J. and Vakharia A.J., 2008. "Heuristics for sourcing from multiple suppliers with alternative quantity discounts", European Journal of Operational Research, **186**, 317-329.
7. Chang C.T, 2006. "An acquisition policy for a single item multi-supplier system with real-world constraints", Applied Mathematical Modeling, **30**, 1-9.
8. Chauhan S.S. and Proth J.M., 2003. "The Concave Cost Supply Problem", European Journal of Operational Research, **148**, 374-383.

9. Hansheng L. and Lishan K., 1999. " Balance between exploration and exploitation in genetic search" , Wuhan University Journal of Natural Sciences, **4 (1)**, 28-32.

10. Hillier F.S and Lieberman G.J. 2005. "Introductions to operations research", McGraw-Hill, **8th ed**, 617-653.

11. Ho W., Xu X. and Dey P.K., 2010. "Multi-criteria decision making approaches for supplier evaluation and selection:A literature review", European Journal of Operational Research, **202,** 16-24.

12. http://www.lindo.com/index.php?option=com_content&view=article&id=34&Itemid=15.

13. Keskin G.A., İlhan S. and Özkan C., 2010. "The Fuzzy ART algorithm: A categorization method for supplier evaluation and selection", Expert Systems with Applications, **37,** 1235-1240.

14. Kheljani J.G., Ghodsypour S.H. and O'Brien C., 2009. "Optimizing whole supply chain benefit versus buyer's benefit through supplier selection", International Journal of Production Economics, **121**, 482-493.

15. Kita E. and Tanie H., 1997. "Shape optimization of continuum structures by genetic algorithm and boundary element method", Engineering Analysis with Boundary Elements, **19,** 129-136.

16. Kumar S., Bagchi T.P. and Sriskandaraja C., 2000. "Lot streaming and scheduling heuristics for m-machine no-wait flowshops", Computers & Industrial Engineering, **38,** 149-172.

17. Liao Z. and Rittscher J., 2007. "A multi-objective supplier selection model under stochastic demand conditions", International Journal of Production Economics, **105**, 150-159.

18. Liao Z. and Rittscher J., 2007. "Integration of supplier selection, procurement lot sizing and carrier selection under dynamic demand conditions", International Journal of Production Economics, **107**, 502-510.

19. Mohammad Ebrahim R., Razmi J. and Haleh H., 2009. "Scatter search algorithm for supplier selection and order lot sizing under multiple price discount environment", Advances in Engineering Software, **40,** 766–776.

20. Qi X., 2007. "Order splitting with multiple capacitated suppliers", European Journal of Operational Research, **178**, 421-432.

21. Reid D.J., 1996. "Genetic Algorithms in Constrained Optimization", Mathematical and computer modeling, **23 (5)**, 87-111.

22. Rosenblatt M.J., Herer Y.T. and Hefter I., 1998. "Note. An acquisition policy for a single item multi-supplier system", Management science, **44**, 96–S100.

23. Saad M., Jones M and James P., 2002. "A review of the progress towards the adoption of supply chain management (SCM) relationships in construction", European Journal of Purchasing & Supply Management, **8**, 173-183.

24. Sahinidis N.V., 2004. "Optimization under uncertainty: state-of-the-art and opportunities" Computers & Chemical Engineering**, 28,** 971-983.

25. Saunders, M.J. 1995. "Chains, pipelines, networks and value stream: the role, nature and value of such metaphors in forming perceptions of the task of purchasing and

supply management", 1st World-wide Research Symposium on Purchasing and supply chain management, Tempe, Arizona, pp.476-85.

26. Sawik T., 2010. "Single vs. multiple objective supplier selection in a make to order environment", Omega, **38 (3-4),** 203-212.
27. Scott C. and Westbrook R., 1991. "New strategic tools for supply chain management". International Journal of Physical Distribution and Logistics, **21 (1)**, 23-33.
28. Tan K.C., Kannan V.R. and Handfield R.B, 1998. "Supply chain management: supplier performance and firm performance", International Journal of Purchasing and Materials Management, **34 (3)**, 2-9.
29. Weber C. and Current J., 1993. "A multi objective approach to vendor selection", European Journal of Operational Research, **68**, 173-184.
30. Xing L., Chen Y., Cai H., 2006. "An intelligent genetic algorithm designed for global optimization of multi-minima functions', Applied Mathematics and Computation, **178**, 355–371.
31. Yimer A.D. and Demirli K., 2010. "A genetic approach to two-phase optimization of dynamic supply chain scheduling". Computers & Industrial Engineering, **58 (3),** 411-422.
32. Youssef H., Sait M.S. and Adiche H., 2001. "Evolutionary algorithms, simulated annealing and tabu search: a comparative study", Engineering Applications of Artificial Intelligence, **14 (2),** 167-181.

33. Yun Y., Moon C. and Kim D., 2009. "Hybrid genetic algorithm with adaptive local search scheme for solving multistage-based supply chain problems", Computers & Industrial Engineering, **56 (3),** 821-838.

Appendix A: Sample medium sized randomly generated problem:

Supplier Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Supplier Price	10	13	7	9	16	20	15	17	40	32	23	16	87	90	26
Min Capacity	354	855	76	214	198	325	292	486	504	571	43	414	121	267	247
Max Capacity	506	891	547	841	814	929	616	831	550	757	568	531	934	569	794

Period number	1	2	3	4	5	6	7	8	9	10	11	12
Demands	5621	1795	6492	2823	7051	7468	8031	4832	907	2494	9987	1653
Allowed Inventory Limit (r)	200	150	100	145	200	100	90	180	256	300	210	150
Inventory Holding Cost for each period (h)	20	20	20	18	20	25	20	40	20	15	20	20

Appendix B: Sample large sized randomly generated problem:

Supplier Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Supplier Price	10	13	7	9	16	20	15	17	40	32	23	16	87	90	26	20	15	17	40	32	23	16	87	90	26
Min Capacity	372	552	531	464	25	698	236	29	50	669	108	342	631	363	69	668	220	413	164	876	199	517	296	467	376
Max Capacity	744	905	618	805	887	979	500	682	522	818	722	660	973	800	825	831	527	657	984	952	738	548	983	701	539

Period number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Demands	12581	3398	2435	18958	3255	622	10649	16725	12751	3643	7074	8793	18676	2983	16276	12234	7157	3642	8126	9142	2298	11278	4337	7340
Allowed Inventory Limit (r)	200	150	100	145	200	100	90	180	256	300	210	150	200	150	100	145	200	100	90	180	256	300	210	150
Inventory Holding Cost for each period (h)	20	20	20	18	20	25	20	40	20	15	20	20	20	20	20	18	20	25	20	40	20	15	20	20